

APPLICATIONS DE LA RÉALITÉ VIRTUELLE

par

Éric Jobidon

mémoire présenté au Département de mathématiques et d'informatique

en vue de l'obtention du grade de maître ès sciences (M. Sc.)

FACULTÉ DES SCIENCES

UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, décembre 1997



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-35687-6

Le 15-04-98 , le jury suivant a accepté ce mémoire dans sa version finale.
date

Président-rapporteur: M. Djemel Ziou _____
Département de mathématiques et d'informatique

Membre: M. Jean-Pierre Dussault _____
Département de mathématiques et d'informatique

Membre: M. Béchir El Ayeb _____
Département de mathématiques et d'informatique

Membre: M. Shengrui Wang _____
Département de mathématiques et d'informatique

Sommaire

Le présent document se veut un rapport de recherche produit suite à la complétion d'une version préliminaire d'un logiciel de navigation dans des environnements virtuels. Le logiciel, nommé *InterWorlds*, offre une technique alternative de recherche à travers de grandes banques de données. L'innovation de celui-ci se situe dans sa capacité de permettre aux utilisateurs de "nager" à travers une information tridimensionnelle plutôt que de consulter les traditionnels écrans bidimensionnels de recherche.

Une description complète du système conçu suit le chapitre expliquant les concepts de la réalité virtuelle. Ensuite, les forces, les faiblesses, les embûches et les solutions envisagées lors de la conception sont présentées. De plus, certaines améliorations non-implémentées sont suggérées. Finalement, une technique de filtrage de signal à l'aide des ondelettes est avancée.

L'aboutissement du long périple que fut la conception de ce logiciel de nouvelle génération s'avère concluant : les critiques reçues lors des expositions sont très encourageantes et, plus que jamais, un besoin persiste dans le créneau des outils de navigation à l'intérieur des grandes banques de données.

Remerciements

L'achèvement de ce document n'aurait pu être rendu possible sans le support, la compréhension, le soutien, les encouragements et les conseils d'individus très influents, chacun à leur façon. Merci à Hélène particulièrement pour son écoute active et son support moral.

Un grand merci aux Drs S. Wang et B. El Ayeb pour leur confiance aveugle ainsi que leurs commentaires constructifs.

Merci à Yves D'aoust pour sa confiance et les multiples possibilités qu'il m'a offertes et finalement merci au FCAR pour un support financier substantiel sans lequel je n'aurais pu poursuivre cet objectif.

Je m'en voudrais de négliger tous les autres qui ont contribué à faire de cette maîtrise un succès : parents, amis, étudiants et autres membres du corps professoral. Un sincère merci à chacun de vous pour votre support tout au long de cette entreprise.

Table des matières

SOMMAIRE	ii
REMERCIEMENTS	iii
TABLE DES MATIÈRES	iv
LISTE DES TABLEAUX	ix
LISTE DES FIGURES	x
INTRODUCTION	1
CHAPITRE 1	
DÉFINITION DE LA RÉALITÉ VIRTUELLE	4
1.1 CATÉGORIES D'UTILISATION	5
1.1.1 <i>Les simulateurs</i>	6
1.1.2 <i>La manipulation à distance</i>	7
1.1.3 <i>La navigation dans un environnement architectural</i>	8
1.1.4 <i>L'aide à la décision</i>	9
1.1.5 <i>Le divertissement</i>	10
1.2 COMPOSITION D'UNE SIMULATION	11
1.2.1 <i>L'application</i>	12
1.2.2 <i>Simulateur d'environnements virtuels</i>	13

1.2.3 Base de données d'objets géométriques tridimensionnels	13
1.2.4 Affichage tridimensionnel	14
1.2.5 Équipement d'entrée/sortie	15
1.2.6 Scénario typique d'interaction	16
1.3 CARACTÉRISTIQUES DES APPLICATIONS	17
1.3.1 Navigation	17
1.3.2 Réalisme de l'environnement	19
1.3.3 Support pour l'audio	21
1.3.4 Support pour les séquences vidéo	22
1.3.5 L'aspect multiusager	23
1.3.5.1 La représentation	23
1.3.5.2 L'interaction	24
1.3.5.3 Les plateformes	25
1.3.5.4 Les mécanismes sous-jacents	26
1.3.6 Description de l'environnement	26

CHAPITRE 2

MÉTHODOLOGIE DU PROJET	28
2.1 PRÉSENTATION SCHÉMATIQUE DU SYSTÈME	28
2.1.1 Diagrammes	29
2.1.2 Hypothèses de base lors de la conception	31
2.1.3 Travaux connexes	36
2.1.3.1 Consultation de séquences vidéo à distance	36
2.1.3.2 Navigation dans les bases de données	38
2.1.3.3 Immersion dans un environnement tridimensionnel	39
2.2 DESCRIPTION TECHNIQUE	40

2.2.1 Le casque d'immersion	41
2.2.2 Le gant numérique	42
2.2.3 InterWorlds Client	43
2.2.4 InterWorlds serveur	45
2.3 DESCRIPTION DE L'ESPACE VIRTUEL	46

CHAPITRE 3

CONCEPTION DU LOGICIEL	50
3.1 PROBLÈMES, DIFFICULTÉS ET SOLUTIONS	50
3.1.1 Sensibilité des capteurs de mouvements	51
3.1.2 Ergonomie des capteurs	54
3.1.3 Calibrage des capteurs	58
3.1.4 Flexion des articulations	61
3.1.5 Orientation du gant	64
3.1.6 Détection des intersections	65
3.1.7 Conversion du signal vidéo	66
3.1.8 Conservation des paramètres	68
3.1.9 Configuration de l'environnement	69
3.1.10 Affichage des séquences vidéo	72
3.1.11 Transmission des données	76
3.1.12 Feed-back tactile	77
3.2 AMÉLIORATIONS FUTURES	78
3.2.1 Conversions de pages HTML en site tridimensionnel	78
3.2.2 L'aspect multiusager	80
3.2.3 Amélioration du réalisme	82
3.2.4 Support pour autres périphériques	83

CHAPITRE 4

FILTRAGE PAR ONDELETTES	85
4.1 INTRODUCTION	85
4.2 APERÇU GÉNÉRAL DES ONDELETTES	86
4.2.1 Définition des ondelettes	87
4.2.2 Types d'ondelettes	88
4.2.2.1 Propriétés des ondelettes	88
4.2.2.2 Exemples d'ondelettes	89
4.2.2.3 Différences entre les types d'ondelettes	91
4.2.3 Analyse par ondelettes	92
4.2.3.1 Transformée ondelette discrète (DWT)	94
4.2.3.2 Transformée ondelette rapide (FWT)	96
4.3 DESCRIPTION INITIALE DU PROBLÈME	98
4.3.1 Description du projet	99
4.3.2 Situation souhaitée	100
4.4 MÉTHODOLOGIE ET PROCESSUS D'IMPLANTATION	101
4.4.1 Description des processus impliqués	101
4.4.2 Intégration du code dans l'application InterWorlds	104
4.4.3 Description des résultats	106
4.5 CONCLUSION	108
CONCLUSION	110
ANNEXES	113
ANNEXE A: INTRODUCTION À LA RÉALITÉ VIRTUELLE	114
ANNEXE B: FILTRAGE PAR ONDELETTES	120

ANNEXE C: L'UPDATEFUNCTION POUR LE GANT	124
ANNEXE D: ABSTRACTION DES ÉTATS DU GANT	126
ANNEXE E: FICHIER DE CONFIGURATION DU MONDE	130
ANNEXE F: EXEMPLE DE FICHIER VRML	133
BIBLIOGRAPHIE	136

Liste des tableaux

Tableau 3.1 : Énumération des états possibles du gant	55
Tableau 3.2 : Association des gestes du gant et des actions	56
Tableau 3.3 : Liste des entrées du Registry	68
Tableau 4.1 : Comparaison de quelques familles d'ondelettes	91

Liste des figures

Figure 1.1 : Schéma de composition d'une simulation	15
Figure 2.1 : Fonctionnement schématique du système	30
Figure 2.2 : Filtrage des capteurs de mouvements	31
Figure 2.3 : Le casque de Virtual i-o	33
Figure 2.4 : Utilisation d'un serveur HTTP	45
Figure 2.5 : L'intérieur de l'environnement	47
Figure 2.6 : L'intérieur d'une aire de visionnage	48
Figure 3.1 : Flexion des articulations dans <i>InterWorlds</i>	63
Figure 3.2 : Représentation du gant dans l'environnement synthétique	64
Figure 3.3 : Différents processus de transmission des séquences vidéo	74
Figure 4.1 : L'ondelette de Haar	90
Figure 4.2 : Algorithme de décomposition et de recomposition FWT	98
Figure 4.3 : Exemples de signaux bruts	100
Figure 4.4 : Exemples d'un signal filtré avec différents paramètres	105

INTRODUCTION

La réalité virtuelle est un concept relativement nouveau qui touche directement à plusieurs disciplines. Bien sûr, l'informatique, les mathématiques et l'électronique y sont omniprésentes, mais des notions de philosophie, d'ergonomie et de sociologie sont tout autant nécessaires pour mettre sur pied une simulation artificielle convaincante.

Le présent document a pour objectif principal la présentation de certains aspects pertinents d'un projet d'élaboration d'un logiciel de navigation dans un environnement tridimensionnel. Ce logiciel, nommé *InterWorlds*, permet encore à ce jour à des utilisateurs, munis de certains périphériques électroniques innovateurs, de plonger dans un monde synthétique représentant un musée et de s'y promener, comme s'ils y étaient véritablement. Tout au long de leur séjour à l'intérieur de cet édifice virtuel, les participants peuvent, à l'aide d'un gant numérique, sélectionner des images ornant la pièce artificielle dans laquelle ils se trouvent et, au contact, une séquence vidéo associée à l'image est automatiquement démarrée. Le projet sera décrit de façon plus approfondie dans le second chapitre, portant sur la méthodologie du projet.

À ce point-ci, il semble opportun de mentionner que ce projet fut réalisé avec les avantages et les contraintes issus de l'encadrement par une firme du milieu privé : Téléglobe Canada, inc. Ainsi, alors que les ressources financières et la visibilité du projet étaient grandes, de rigoureux délais et des exigences matérielles furent imposés.

L'objectif principal de ce document est la décomposition complète du logiciel *InterWorlds* en concepts théoriques reconnus par la communauté scientifique. Il sera question des différents

modules composant une telle application d'immersion et de quelques facettes permettant d'apprécier de tels efforts de développement.

La littérature scientifique actuelle ne présente aucune recherche traitant de l'apport de la réalité virtuelle sur la navigation à l'intérieur de larges banques de données. Certes, plusieurs auteurs s'intéressent au domaine de la réalité virtuelle et d'autres, à la navigation à travers des bases de données, mais jamais la combinaison de ces deux disciplines n'avait fait l'objet d'un article de publication. Malgré cette évidente lacune, le lecteur retrouvera, tout au long du document, certaines références tirées de ces travaux connexes.

Chacune à sa façon, ces références apportent un enrichissement complémentaire à la lecture de ce mémoire et permettent de tisser des liens entre les multiples disciplines intervenant dans la réalité virtuelle.

Dans le chapitre 1, une couverture sommaire de quelques aspects théoriques de la réalité virtuelle sera effectuée. Tout au long de ce chapitre, les aspects arides de ce nouveau concept seront illustrés par des facettes du projet réalisé et celles-ci serviront aussi à préparer l'argumentation utilisée pour justifier les avenues choisies lors du développement.

Ce n'est qu'une fois la base conceptuelle bien établie que la description de la méthodologie utilisée sera étalée. Dans ce second chapitre, une description des entités d'*InterWorlds* ainsi que du développement de ce logiciel sera effectuée. Dans ce même chapitre, certaines images du produit fini ainsi qu'une description complète de l'environnement virtuel créé seront révélées. Puis, dans le chapitre 3 figurera un résumé des embûches rencontrées, des solutions étudiées ainsi que des conclusions finalement retenues.

Tous les aspects ayant trait au domaine fascinant des ondelettes seront abordés à l'intérieur du chapitre 4. Les concepts théoriques combinés à quelques exemples permettront d'illustrer comment un tel concept a été utilisé dans le cadre de ce projet.

Finalement, suite à l'analyse poussée des aspects pertinents du projet, des conclusions seront tirées sur l'achèvement de celui-ci ainsi que sur son avenir.

CHAPITRE 1

Définition de la réalité virtuelle

Les immenses et attrayantes possibilités qu'offre la réalité virtuelle en font une discipline pleine d'avenir. Dans les prochaines sections, une définition du concept ainsi que les composants logiques de celui-ci seront énumérés. Puis, suivra un bref aperçu des caractéristiques des applications de réalité virtuelle¹.

La terminologie **réalité virtuelle** est, de nos jours, un concept avant-gardiste qui est de plus en plus utilisé. En fait, son utilisation est telle que le terme lui-même ne signifie plus grand chose tellement il est utilisé à toutes les sauces. Une définition de la réalité virtuelle est :

«Un terme générique associé aux systèmes informatiques créant une expérience visuelle / auditive / d'immersion en temps réel» [1].

Cette définition, très générale pourrait judicieusement être précisée de la façon suivante : *«la réalité virtuelle, c'est une interface simulant un environnement dans lequel un utilisateur peut plus facilement manipuler des données»*.

¹ Le lecteur trouvera, en annexe A, un document supplémentaire vulgarisant certains des concepts reliés à cette nouvelle discipline qu'est la réalité virtuelle.

Dans une telle définition, le concept de donnée réfère à une parcelle d'information quelconque, qu'elle soit de nature textuelle, audio, vidéo ou de nature plutôt abstraite. Tout en demeurant encore relativement générale, cette définition laisse transparaître que la réalité virtuelle est une vitrine sur un (ou à l'occasion plusieurs) environnement synthétique, le plus souvent tridimensionnel, dans lequel le ou les participants peuvent interagir avec certains éléments, certaines informations. La majorité des belligérants sérieux du domaine s'entendent sur cette dernière définition [2].

Au-delà de toute la frénésie entourant la terminologie, la réalité virtuelle d'aujourd'hui est constituée d'un ou de plusieurs participants interagissant les uns avec les autres, ainsi qu'avec le reste de l'environnement, en utilisant des périphériques d'immersion gérés par un certain nombre de puissants ordinateurs. Les prochains paragraphes serviront à préciser ces quelques concepts.

1.1 Catégories d'utilisation

Alors qu'*InterWorlds*, le projet dont il est question dans le présent document, utilise la réalité virtuelle pour plonger les participants dans un environnement calquant notre réalité, plusieurs types alternatifs d'expérience d'immersion virtuelle existent. En effet, le monde d'immersion peut être inspiré d'un endroit physique réel ou non. Par exemple, alors que certains mondes virtuels représentent une construction architecturale particulière, d'autres tenteront plutôt de placer l'utilisateur dans un environnement purement imaginatif, tel qu'une cité futuriste. En plus de ces deux types d'environnements, il est aussi possible de décomposer ce nouveau média en fonction de certaines catégories d'utilisation. À cet effet, voici une liste sommaire

des principaux domaines d'application de la réalité virtuelle ainsi que les caractéristiques déterminantes de chacun.

1.1.1 Les simulateurs

Cette première catégorie d'utilisation regroupe, sans y être limité, autant les simulateurs de vol, les simulateurs de course, que ceux permettant à un militaire de parfaire son entraînement en le plongeant dans un milieu tridimensionnel synthétique. Le but premier de cette catégorie d'utilisation est d'immerger le participant dans une copie quasi parfaite d'un environnement très particulier afin de lui inculquer ou de lui faire pratiquer certaines techniques relatives à son domaine d'expertise.

Cet aspect de haute conformité de la simulation avec une situation réelle introduit indubitablement une contrainte importante quant à la qualité du réalisme de l'environnement. En effet, pour produire une simulation efficace, cette dernière doit plonger le participant dans un milieu synthétique correspondant le plus possible avec ce qui l'attend à l'extérieur du simulateur, une fois l'entraînement terminé. Ainsi, pour permettre à l'utilisateur de pouvoir interagir de façon concrète avec son environnement, les concepteurs de ce type d'expérience d'immersion ont souvent recours à l'utilisation d'objets physiques palpables pour augmenter le réalisme de l'entraînement puis, du même coup, le degré d'immersion d'une simulation. Avec le casque d'immersion et le gant numérique habituellement utilisés en réalité virtuelle, il est impensable de pouvoir imiter le comportement, les détails et surtout la complexité d'une cabine de pilotage d'un avion de ligne, par exemple. De telles d'expériences virtuelles sont presque exclusivement augmentées à l'aide des objets composant l'environnement de travail immédiat du participant, lorsque qu'il se retrouvera dans une situation similaire, à l'extérieur du simulateur. Qu'il s'agisse d'une cabine de pilotage, d'un habitacle de bolide de course ou d'un terrain marécageux, le réalisme de ces environnements est critique au succès de ce type

d'expérience virtuelle, en faisant ainsi une caractéristique distinctive de cette catégorie de mondes virtuels.

1.1.2 La manipulation à distance

Un autre type d'expérience virtuelle consiste à contrôler un objet à distance. Ce genre d'application est particulièrement important dans des situations où les objets en question sont inatteignables de par leur taille, leur proximité géographique ou simplement par mesure de sécurité pour les experts. Parmi les utilisations de la réalité virtuelle correspondant à cette catégorie, les plus connues sont :

- la **télémédecine**, ou la pratique d'interventions chirurgicales sur des patients physiquement éloignés des praticiens médicaux (incisions, ablations, etc. effectuées par des robots contrôlés par des experts à distance) [3][4],
- le **pilotage de robots ou de sondes à distance**, alors que les experts demeurent dans les laboratoires, des sondes, tantôt sous-marines, tantôt situées sur d'autres planètes, leur permettent d'effectuer de l'exploration de mondes réels autrement hors de portée [5],
- la **manipulation moléculaire**, où les chercheurs ont la possibilité de naviguer à travers un environnement microscopique géant et d'interagir avec les molécules en saisissant et en manipulant leur représentation tridimensionnelle dans l'environnement synthétique, avec le gant numérique [6],

Très souvent, dans ce type de simulation, les participants utilisent la réalité virtuelle pour manipuler un objet réel. Afin d'y parvenir, un signal vidéo provenant de l'objet contrôlé doit leur être transmis avec un minimum de délai, ceci afin de permettre aux experts de déplacer l'objet en question le plus fidèlement possible, en fonction de l'environnement externe dans

lequel cet objet se retrouve. C'est le plus souvent une forme de rétroaction visuelle qui permet aux experts de voir les résultats de leur intervention.

1.1.3 La navigation dans un environnement architectural

Cette troisième tendance du domaine regroupe des applications telles que le logiciel permettant de se retrouver virtuellement dans le stade olympique d'Atlanta² avant même que sa construction soit complétée, les outils utilisés par les grandes firmes d'architecture pour permettre à leurs clients de visiter un édifice alors que celui-ci est encore au stade conceptuel, etc.

Étant donné le haut degré de complexité et les exigences du côté de la qualité du réalisme de telles expériences, il est facile d'imaginer que pour pouvoir exécuter de telles simulations, la capacité de traitement des ordinateurs hôtes doit être excessivement élevée. En effet, si, en plus d'un faible réalisme, la machine peu puissante ne permet que l'affichage de cinq ou dix images par seconde, l'expérience deviendra trop peu convaincante et beaucoup trop saccadée pour en tirer quoi que ce soit, hormis, peut-être, un mal de tête et une forte nausée³.

Les logiciels d'immersion qui correspondent à cette catégorie possèdent beaucoup de points en commun avec la première catégorie, celle des simulateurs. En effet, la qualité du réalisme et la rapidité d'exécution sont dans les deux cas des aspects distinctifs de ces classes. Par contre, la distinction réside dans le fait que pour les simulateurs, l'augmentation de l'immersion à l'aide d'objets réels (tel un cockpit ou les tableaux de bord d'un avion de chasse), est requise alors qu'elle est loin d'être nécessaire pour la navigation architecturale.

² Un tel logiciel fut mis au point par la multinationale IBM, pour promouvoir son apport pour les jeux d'Atlanta, en 1996.

³ Se référer aux sections 2.1.2 et 2.2.1 pour de plus amples détails sur, respectivement, le seuil du nombre d'images par seconde et pour le réflexe vestibulo-oculaire.

Le logiciel *InterWorlds*, dont il sera question à l'intérieur de ce document, illustre aussi très bien certains aspects de cette catégorie. La qualité du graphique et la vitesse d'affichage y sont, tout autant pour *InterWorlds*, importantes. Cet aspect sera couvert plus en profondeur dans le second chapitre.

1.1.4 L'aide à la décision

Les outils d'aide à la décision sont de plus en plus populaires sur le marché. Ils permettent aux cadres de facilement gérer d'importantes masses d'information de façon visuelle et intuitive. La transformation de ces logiciels en applications de réalité virtuelle permet de plonger l'utilisateur à même les données qu'il manipule. Ainsi, le courtier en bourse peut surveiller en temps réel le cours de son portefeuille d'actions simplement en regardant des prismes de différentes couleur monter et descendre autour de lui, dans un environnement synthétique. De la même façon, le gestionnaire de projets obtient de l'information quant à l'attribution des ressources pour un certain projet en cours en observant une représentation graphique tridimensionnelle de l'information dans laquelle il peut se déplacer.

Ce genre d'application est caractérisé par un faible réalisme car un haut niveau de qualité graphique est souvent superflu pour ce type d'immersion. L'environnement constitue une représentation tridimensionnelle de l'information à traiter et cette représentation est plus souvent qu'autrement abstraite. Par contre, un aspect très important est la correspondance entre les données réelles et leur représentation dans l'environnement. La mise à jour de l'information doit souvent se faire en temps réel pour permettre aux dirigeants de prendre des décisions rapides avec des données fidèles.

L'aspect de l'utilisation relève aussi une autre caractéristique spécifique: la facilité de l'utilisation. En effet, étant donnée la nature de la clientèle type visée par ce genre d'outil de

travail, la navigation, la manipulation et la prise de décision doivent être simples et surtout très intuitives. Les cadres n'ayant souvent pas le temps d'apprendre à utiliser un outil trop compliqué, celui-ci doit être facile à utiliser et apporter des avantages évidents faute de quoi les utilisateurs ne seront pas enclins à changer d'outil de travail.

En ce sens, *InterWorlds* se rapproche aussi de cette catégorie d'application car il consiste en un outil de navigation dans d'importantes banques de données. Les données, en l'occurrence des séquences vidéo, ont une représentation concrète dans l'environnement et les participants doivent pouvoir y naviguer avec aisance et prendre la décision de déclencher un visionnage sans difficulté. De plus, la navigation à l'aide du gant numérique doit être simplifiée le plus possible car elle constitue tout de même un nouveau moyen de communication avec la machine et, comme pour la souris lors de son introduction, son utilisation n'est pas innée et ne doit être apprise que graduellement.

1.1.5 Le divertissement

Cette dernière catégorie principale d'utilisation de la réalité virtuelle est, à ce jour, probablement la plus développée de toutes, en termes de marché. C'est en effet à travers le divertissement que la réalité virtuelle s'est fait connaître aux yeux de plusieurs et c'est dans des industries en plein essor comme les arcades et les jeux vidéo qu'une grande partie des investissements dans des projets de réalité virtuelle se retrouve [7].

La catégorie de la réalité virtuelle appliquée au divertissement, tout comme la majorité des jeux vidéo, requiert elle aussi une machine relativement rapide, afin de permettre une immersion convaincante et une interaction instantanée dans un environnement passablement complexe. Le monde synthétique est souvent imaginatif et artistique, mais la qualité de celui-ci doit être tout de même élevée pour satisfaire aux exigences grandissantes des joueurs.

Les applications de divertissement sont souvent caractérisées par un contenu distrayant et attirant l'attention par les couleurs, l'animation, etc. Étant donné son coût abordable, c'est bien entendu cette catégorie qui est la plus populaire et qui permet de mieux faire connaître la réalité virtuelle.

D'autres catégories, celles-là moins importantes ou moins riches du côté immersion, existent, mais la liste ci-haut, sans se vouloir exhaustive, permet de facilement classer les applications de réalité virtuelle quant à leur contenu. Mais hormis ces cinq grandes tendances de la réalité virtuelle, il semble aussi intéressant de décomposer une application d'immersion dans un monde synthétique en modules distincts, effectuant chacun une tâche précise, essentielle au fonctionnement adéquat du logiciel.

À cet effet, la majorité des logiciels de réalité virtuelle sont constitués de cinq principaux modules de base. Chacun de ceux-ci sera expliqué dans la section suivante puis, dans le prochain chapitre, un parallèle sera effectué avec *InterWorlds* et les différents composants qu'il intègre.

1.2 Composition d'une simulation

La conception d'un logiciel d'immersion dans des mondes virtuels est une tâche complexe et pour y parvenir il semble approprié de décomposer l'application en modules distincts, chacun accomplissant une activité particulière⁴.

⁴ Cette décomposition de base d'un logiciel d'immersion en modules est tirée de [7].

1.2.1 L'application⁵

Le premier des cinq modules de base qui composent un logiciel de navigation dans un environnement virtuel est celui qui effectue la gestion de tous les autres modules : c'est l'application. En effet, c'est cette entité qui détermine :

- les comportements des objets à l'intérieur de la simulation,
- la modification des comportements des objets de l'environnement,
- les types d'interactions possibles,
- la possibilité ou non de partager l'environnement avec d'autres utilisateurs,
- la possibilité ou non d'ajouter ou de soustraire dynamiquement des objets dans la simulation,
- etc.

C'est aussi l'application qui permet de configurer et de paramétrer le fonctionnement des divers modules. Le participant, à travers ce module, peut ajouter des objets à la simulation, modifier les caractéristiques et même les comportements de ceux-ci.

Dans une simulation où plusieurs participants peuvent interagir, il faut prévoir un ensemble de règles régissant cette configuration faute de quoi des problèmes pourraient survenir. En effet,

⁵ Dans [7], il porte le nom de *User Application* ou *APPL*.

si plusieurs utilisateurs entreprennent des modifications qui sont paradoxales entre elles, c'est le module application qui déterminera comment les gérer et surtout, comment répercuter ces changements à travers toutes les copies du monde.

1.2.2 Simulateur d'environnements virtuels⁶

Le VWS supporte l'application et est composé de deux niveaux : la simulation des objets et la simulation générale. La simulation générale fonctionne en tout temps et est responsable du positionnement de tous les objets, des translations de ceux-ci, de la détection des collisions, etc. et ce pour tous les objets dans une région sous son contrôle.

Cette décomposition supplémentaire, accomplie en subdivisant ce module en deux entités plus spécifiques, permet un plus haut niveau de contrôle sur l'interaction de chaque objet avec le reste de l'environnement. Alors que le simulateur des objets intervient à une échelle microscopique, soit au niveau de chaque objet, la simulation générale veille à un niveau plus élevé, par exemple lorsqu'un déplacement fait intervenir plus d'un objet.

1.2.3 Base de données d'objets géométriques tridimensionnels⁷

Le troisième module de base composant un logiciel d'immersion est celui contenant la représentation géométrique de tous les objets de la simulation. C'est donc à l'intérieur de cette banque d'informations que les coordonnées, la couleur, la texture et les propriétés physiques de chaque objet se retrouvent. De plus, lorsque c'est le cas, cette entité contient

⁶ Originellement nommé dans [7] : *Virtual World Simulator* ou *VWS*.

⁷ Dans [7], le module se nomme *3D Geometric Database* ou *3DDB*.

aussi le script ou la définition du comportement et des actions de l'objet dans l'environnement.

Par exemple, si l'objet en question est une porte, la couleur, le type de matériau de construction la composant, les coordonnées de celle-ci et une foule d'autres informations pertinentes y seront stockées. De plus, s'il est jugé pertinent que la porte "ouvre" lorsqu'un participant la touche dans l'environnement synthétique, un script ou une définition quelconque du mouvement à effectuer suivant ce contact y est aussi conservé.

1.2.4 Affichage tridimensionnel⁸

Une fois les objets, représentés dans la base de données, mus et contrôlés par le module de simulation (VWS), il est nécessaire d'informer le participant du résultat de ces déplacements sur le reste de l'environnement. C'est à ce point-ci que la quatrième entité intervient. Le module d'affichage tridimensionnel traduit les objets, les sons et les contacts physiques dans une représentation adéquate à l'intérieur du monde virtuel. Pour accomplir cette tâche, le 3DR obtient du 3DDB les propriétés physiques des objets de l'environnement et les traduit en couleurs, en jeux de lumière, en intensités sonores ou autres entités sensorielles pour finalement les faire parvenir au participant.

Puis, une fois l'état de l'environnement traduit dans son format final, toute l'information est transférée au prochain module, celui effectuant l'acheminement des informations vers les périphériques de sortie.

⁸ Traduit de *3D Rendering* ou *3DR* dans [7].

1.2.5 Équipement d'entrée/sortie⁹

C'est à travers les périphériques disponibles que l'ordinateur parvient à "communiquer" avec l'utilisateur. Qu'il s'agisse d'informations calculées par la machine et destinées au participant via un casque, un écran ou des haut-parleurs ou, inversement, de stimuli provenant des capteurs de mouvements¹⁰ étant destinés à l'ordinateur, ce dialogue s'effectue à travers le module d'entrée/sortie. La figure 1.1 illustre schématiquement comment cet échange d'information s'effectue.

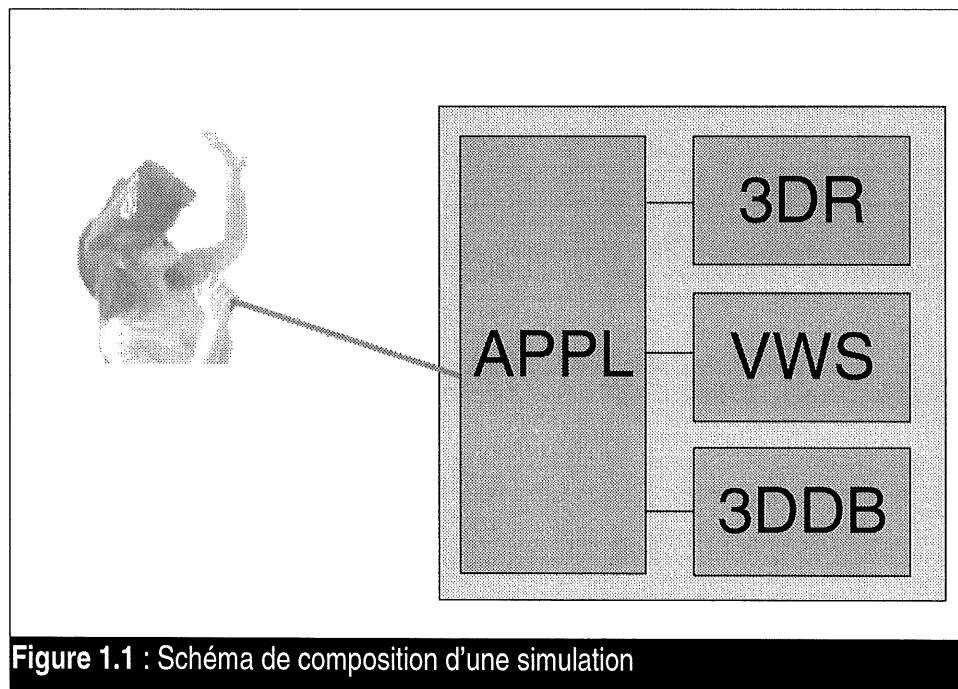


Figure 1.1 : Schéma de composition d'une simulation

⁹ Identifié par la terminologie *I/O Devices* ou *I/O* dans [7].

¹⁰ Aux sections 3.1 et suivantes, le fonctionnement des divers périphériques utilisés dans le cadre du projet *InterWorlds* est décrit plus en profondeur.

1.2.6 Scénario typique d'interaction

Les prochains paragraphes visent à illustrer le processus d'interaction entre les cinq modules définis plus haut. Il s'agit bien entendu d'un exemple fictif créé dans le simple but de concrétiser la théorie rapidement survolée dans cette section.

À cet effet, le lecteur est fortement invité à consulter les documents mentionnés dans la bibliographie, particulièrement ceux retrouvés dans [7], pour approfondir les notions aperçues dans les sections précédentes.

Dans cet exemple, la simulation est constituée d'un environnement synthétique micro-cellulaire où le participant peut s'y déplacer sous la forme d'un globule rouge circulant dans le sang. Malgré le nombre démesuré de globules dans le sang, une seule copie de l'objet *globule* est retrouvée dans la base de données d'objets géométriques tridimensionnels. L'information qui s'y retrouve est partagée par toutes les copies de l'objet puisque chacune de celles-ci possède les mêmes caractéristiques physiques que l'original, stocké dans le 3DDB. Puis, à chaque cycle de la simulation, le simulateur d'environnements virtuels calcule le déplacement des globules clonés selon les lois en vigueur dans l'environnement (gravité, friction, etc.). Ensuite, le VWS envoie l'état de la simulation sous forme de commandes au module d'affichage tridimensionnel, qui se charge alors de générer l'image¹¹ finale pour l'affichage à l'utilisateur. La représentation ultime est finalement acheminée au module d'entrée/sortie pour ensuite aboutir dans les périphériques de sortie.

Toute l'interaction à l'intérieur de ce scénario doit s'effectuer dans une période de temps très courte pour ne pas donner au participant l'effet d'assister à un diaporama ou, pis encore, pour ne pas lui faire subir le mal des simulateurs ainsi que les nausées s'y rattachant. En effet, le

¹¹ Il s'agit ici autant de l'image visuelle, auditive, gustative qu'olfactive, lorsque la technologie le permet.

tout doit s'effectuer en moins de 3/10e de seconde¹² pour éviter tout dérèglement du réflexe vestibulo-oculaire¹³.

1.3 Caractéristiques des applications

Puisqu'il existe plusieurs types d'expériences virtuelles, il importe de pouvoir classifier celles-ci selon certains critères prédéterminés. Ces principales caractéristiques internes constituent en fait des traits désirables pour des logiciels de réalité virtuelle. De plus, elles permettent de mieux apprécier l'immersion proposée par chacun, selon un point de vue plus objectif.

Dans les prochains paragraphes, il sera question de quelques-uns de ces critères. Chacun sera expliqué, puis un parallèle sera effectué entre cette théorie et le logiciel *InterWorlds* développé dans le cadre de ce projet.

1.3.1 Navigation

Le premier de ces six critères est la navigation. C'est la navigation dans un environnement synthétique qui donne l'effet au participant de se déplacer à l'intérieur d'un autre monde. Cet effet de liberté de déplacement n'est accompli, en fait, qu'en déplaçant les entités de la simulation dans la même amplitude que le mouvement de l'utilisateur, mais simplement dans

¹² Selon [8].

¹³ Se référer à la section 2.2.1 pour plus de détails à ce sujet.

le sens inverse. En altérant ainsi la position des entités solides, le participant a, jusqu'à un certain degré, l'impression de se mouvoir dans le monde artificiel.

Certains aspects de cette navigation sont critiques au réalisme de celle-ci. Par exemple, si "l'intuitivité" du mouvement est limitée par des périphériques restrictifs comme une manette de jeu, l'immersion dans l'environnement sera moins convaincante que si le participant peut physiquement déplacer ses jambes pour avancer. De façon similaire, un autre aspect est la quantité de déplacement qui est produite lorsqu'une telle action est déclenchée. Si l'utilisateur avance trop vite ou trop lentement, le réalisme en souffrira grandement.

Un autre de ces aspects réside dans la qualité des périphériques supportés pour effectuer cette navigation. Ces derniers sont-ils ergonomiques et faciles à utiliser? Sont-ils restrictifs dans les mouvements supportés? Dans quelle mesure? Outre la qualité, la quantité est tout autant critique. Si trop peu de capteurs de mouvements sont disponibles, des mouvements intuitifs deviennent impossibles. De même, si trop de capteurs sont utilisés, un utilisateur occasionnel risque de se sentir perdu à travers la multitude de mouvements possibles alors qu'un habitué pourra s'y déplacer plus aisément. Dans le cas d'*InterWorlds*, puisque l'utilisateur type est un spectateur peu familier avec de telles expériences, seulement deux périphériques d'entrée/sortie ont été utilisés : un gant numérique pour les déplacements et un casque d'immersion pour les orientations. Les détails techniques ainsi que l'utilisation de chacun seront avancés dans le prochain chapitre.

Il est toutefois opportun, à ce point, de mentionner que le déplacement dans un tel environnement assuré par un simple gant numérique s'est avéré plus ou moins convaincant. Peut-être la qualité de ce dernier est aussi à blâmer, mais il semble que son utilisation s'apparente à celle de la souris lors de son introduction il y a une dizaine d'années : les participants doivent s'y habituer et les premières minutes de la simulation ne servent en fait qu'à titre de période d'apprentissage pour l'utilisation de ce nouveau périphérique.

La faible intuitivité de ce moyen de déplacement se traduit aussi par une incapacité de doser l'amplitude du mouvement. Puisqu'un seul mouvement lui permet d'avancer, il lui est impossible de préciser un déplacement plus rapide pour courir dans l'environnement, ou plus lent pour y marcher.

Certains aspects auraient avantage à être abordés dans une version ultérieure du logiciel pour ainsi augmenter le degré de réalisme de la simulation.

1.3.2 Réalisme de l'environnement

La seconde caractéristique des appareils d'immersion est le degré de réalisme de l'environnement tridimensionnel. À l'intérieur d'une simulation, le sens le plus souvent stimulé est bien évidemment la vue. Pour cette raison, il est très important d'augmenter la qualité de l'affichage de tous les objets de la simulation, dans la mesure du possible. Par contre, dans les faits, avec les limitations matérielles omniprésentes, il est encore aujourd'hui impossible d'arriver à afficher une séquence animée avec une qualité d'image approchant le photo-réalisme, à une cadence respectable¹⁴.

Un certain compromis peut être atteint en concentrant les efforts d'affichage sur les objets se retrouvant à l'intérieur du champ de vision immédiat du participant. Ainsi, tous les objets se situant à moins de dix "unités virtuelles" (peu importe si celles-ci correspondent à des mètres, à des pouces ou à une autre unité de mesure) du participant se verront accorder cinquante pour cent du temps de traitement réservé à l'affichage par exemple, alors que les entités à l'extérieur de ce rayon se partageront le reste des efforts d'affichage.

¹⁴ Le terme "respectable" est volontairement ambigu et permet ici d'alléger le texte d'une discussion qui sera abordée à la section 2.2.1.

En plus de l'aspect externe des objets d'une simulation, la nature statique ou dynamique de ceux-ci influe aussi grandement sur le réalisme visuel d'une scène. En effet, si, dans un monde synthétique, les portes peuvent être ouvertes puis refermées plutôt que d'être fixes, si un certain objet peut être déplacé ou peut même se déplacer par lui-même, la sensation d'immersion qu'éprouvera l'utilisateur sera immanquablement décuplée et ce dernier sera beaucoup plus enclin à prolonger son "séjour virtuel" dans un tel environnement. Cette capacité d'interaction que certains logiciels permettent, qui peut se traduire par l'activation d'interrupteurs ou par la rotation de poignées de portes, augmente le réalisme d'une immersion et, par le fait même, donne un caractère plus convaincant à l'expérience.

InterWorlds, de son côté, ne parvient pas au niveau de photo-réalisme que d'autres logiciels atteignent. En fait, l'environnement statique subit un traitement préparatoire avant d'être intégré à la simulation. Cette étape, consistant à calculer les ombres et les jeux de lumière de chaque objet de l'environnement, permet de passer outre l'aspect de blocs et de figures géométriques simples qu'a la majorité des logiciels de réalité virtuelle. Quoique conservant un aspect graphique plutôt que photo-réaliste, cet effet augmente la crédibilité d'un environnement de façon significative.

Dans *InterWorlds*, hormis le participant lui-même, aucun objet ne peut être déplacé dans l'environnement. Par contre, des images sur les murs virtuels, agissant à titre de liens hypertextes, peuvent être sélectionnées pour automatiquement déclencher le visionnage d'une séquence animée. Cette possibilité d'interaction que le logiciel permet est très importante et, grâce à elle, les participants considèrent leur expérience virtuelle plus intéressante et réelle, dans la mesure où elle leur rappelle un environnement physique concret augmenté à l'aide de fonctions de consultation interactive.

1.3.3 Support pour l'audio

Le second sens le plus stimulé dans une expérience d'immersion virtuelle est l'ouïe. Puisque les stimulations auditives sont omniprésentes dans la vie de tous les jours, il est normal de retrouver des sons ou des bruits à l'intérieur de l'environnement synthétique. Qu'il s'agisse de musique, de parole ou d'effets spéciaux (tel que le bruit des pas d'un autre participant qui s'approche), la présence de sons dans une simulation augmente grandement la qualité de l'immersion.

Dans *InterWorlds*, étant donné qu'un seul utilisateur à la fois peut interagir dans l'environnement, les sons provenant d'autres participants sont inexistantes. Il était donc important de fournir d'autres stimulations auditives pour pallier à cette lacune. Il fut donc choisi d'incorporer une musique de fond omniprésente dans l'environnement, sans que celle-ci soit nuisante lors de la consultation de séquences vidéo. En effet, si, à cause de la trame sonore en arrière plan, le participant ne parvient pas à entendre le contenu audio de la séquence animée qu'il a choisi, la musique de fond devient nuisante.

Dans une version ultérieure d'*InterWorlds*, il pourrait s'avérer intéressant d'associer une séquence audio descriptive à chaque image de l'environnement pouvant être sélectionnée. L'utilisateur pourrait ainsi bénéficier d'un indice auditif quant à la nature du contenu associé à une image dans le monde virtuel. Simplement en s'approchant d'une image, la description pourrait automatiquement se faire entendre, de façon similaire à ce qu'un guide ferait dans un musée. Avec un tel mécanisme, la navigation serait facilitée et la sélection des séquences à consulter, beaucoup plus aisée.

Le support pour le son tridimensionnel serait aussi bénéfique. Cette technologie permet à un participant de facilement déterminer le point précis de l'espace d'où provient un son

quelconque. Une telle technique commence à peine à être utilisée, mais s'avère déjà très prometteuse [9].

1.3.4 Support pour les séquences vidéo

Afin que l'immersion virtuelle devienne une expérience multimédia enrichissante, il est souvent approprié d'y inclure des capacités de visionnage pour les séquences vidéo. Tel que le support des effets sonores, décrit dans la section précédente, les séquences animées rendent un monde virtuel plus intéressant et plus riche. En fait, qu'il s'agisse de séquences animées, de films ou même de vidéoconférences, un tel ajout permet à l'utilisateur de constater à quel point la valeur ajoutée de ce type d'expérience d'immersion est imposante. Même si l'environnement virtuel représente une scène réelle, comme un musée par exemple, la disponibilité de quelques animations ou de films descriptifs expliquant une œuvre exposée ajoute énormément à la visite du musée virtuel par opposition à une visite **réelle**, tant au plan informatif qu'au niveau ludique de l'expérience.

De par sa nature, *InterWorlds* intègre aussi des mécanismes pour le visionnage de telles séquences. Puisque la base de données d'*InterWorlds* est constituée exclusivement de séquences vidéo, certaines avenues permettant l'affichage de telles animations furent étudiées. La solution retenue permet la superposition d'un clip vidéo MPEG¹⁵ sur le monde virtuel. Une description plus complète des alternatives peut être retrouvée à la section 3.1.10.

¹⁵ De l'anglais **M**otion **P**icture **E**xpert **G**roup. C'est une norme d'encodage et de compression d'un signal vidéo analogique.

1.3.5 L'aspect multiusager

De par sa nature, la réalité pousse à l'interaction entre les différents intervenants. Il est donc logique de poursuivre l'évolution des environnements virtuels jusqu'à permettre à plusieurs utilisateurs d'interagir à l'intérieur d'un même environnement virtuel. Certainement l'un des aspects les plus susceptibles d'augmenter le degré d'immersion d'un logiciel de réalité virtuelle est ce support pour plusieurs utilisateurs. Cette facette d'une application d'immersion peut aussi être subdivisée en quatre éléments distincts : la représentation, l'interaction, la plateforme et les mécanismes sous-jacents. Puisque le logiciel *InterWorlds* ne permet pas, à ce point-ci, de représenter plusieurs participants dans l'environnement, il sera aussi question, tout au long de cette section, des scénarios envisageables pour chacun de ces éléments. Ces nouvelles fonctionnalités pourraient certainement être incorporées dans une prochaine version du logiciel.

1.3.5.1 La représentation

La première de ces facettes regroupe les différentes caractéristiques relatives à la représentation d'un individu à l'intérieur de l'environnement. À cet effet, certains logiciels optent pour une technique favorisant la rapidité d'affichage et d'exécution de la simulation en substituant une image bidimensionnelle très simple à chaque individu. D'autres opteront plutôt pour une représentation par un modèle tridimensionnel quelconque, qu'il soit réaliste ou non. Par exemple, un individu de l'environnement pourrait se voir traduit par un ensemble simpliste de six cubes agencés de façon à simuler une forme humaine ou, alternativement, dans le cas d'un utilisateur plus fortuné ayant accès à une machine avec des capacités graphiques plus avancées, par une forme plus complexe comme un dragon composé de milliers de polygones dans certains cas.

Il pourrait aussi être intéressant d'augmenter l'effet de réalisme en ornant un modèle tridimensionnel avec une photographie de l'utilisateur, ou même avec une séquence animée correspondant, en temps réel, aux expressions de celui-ci. Ceci pourrait aussi être accompli en captant le visage de chaque participant en utilisant une caméra numérique et en affichant ce film sur la tête du modèle tridimensionnel représentant l'utilisateur dans l'environnement synthétique.

1.3.5.2 L'interaction

La présence d'autres individus dans un monde tridimensionnel artificiel entraîne automatiquement l'ajout d'une nouvelle vocation à l'environnement : l'interaction. Cette nouvelle possibilité d'interaction avec d'autres participants semble provoquer une abstraction systématique de l'aspect artificiel du monde pour la grande majorité des utilisateurs. En effet, l'aspect artificiel de l'environnement est rapidement oublié alors que l'utilisateur devient psychologiquement absorbé par son interaction avec les autres participants. À un certain point, la capacité d'interaction devient plus importante que la qualité graphique des images de l'environnement [10].

Cette interaction peut toutefois prendre plusieurs formes. Alors que dans la vie quotidienne les humains concentrent leurs contacts sous des formes auditives et visuelles, un environnement artificiel, avec ses contraintes et ses avantages, pourrait ouvrir la porte à de nouvelles méthodes favorisant les échanges avec autrui. Quoique le calquage des méthodes *traditionnelles* d'interaction puisse aussi s'avérer approprié, il serait fort probablement très intéressant de faciliter l'émergence de nouvelles techniques de communication en ajoutant le support pour quelques autres périphériques d'entrée/sortie. Peut-être deviendrait-il alors plus naturel, lors d'une rencontre fortuite avec un autre participant, de communiquer par la musique plutôt que d'échanger le traditionnel : "Bonjour!" [11].

L'aspect important à retenir est de minimiser les restrictions imposées au participant tout au long de son expérience virtuelle. Si celui-ci désire communiquer avec chacun de ses sens, le logiciel devrait le lui permettre le plus facilement possible. Le degré d'immersion ne s'en trouverait que décuplé et l'expérience de l'utilisateur, améliorée.

Certes, des contraintes technologiques empêchent présentement les logiciels d'intervenir sur chacun des sens, mais aussitôt qu'un périphérique permettant de reproduire les odeurs de façon automatique sera disponible, il sera primordial de l'intégrer à un logiciel d'immersion afin de permettre aux utilisateurs de communiquer et d'interagir d'une façon entièrement nouvelle.

1.3.5.3 Les plateformes

Alors que les deux premiers éléments ont plutôt trait à la sémantique du support pour plusieurs participants, les deux suivants relèvent surtout de l'aspect syntaxique de cette situation.

Hormis la représentation des individus et les types d'interaction qu'il peut survenir entre chacun, il importe aussi de se pencher sur la nature et le nombre de plateformes qui sont supportées par le logiciel. Il existe, ici aussi, quelques variables permettant de catégoriser les plateformes supportées. À cet effet, les principales sont : la puissance de la machine, son coût et le taux relatif d'utilisation (c'est-à-dire son indice de pénétration sur le marché).

Bien entendu, le principal objectif est de ne pas restreindre le nombre de participants en supportant le plus grand nombre de plateformes. Qu'il s'agisse d'un ordinateur personnel, de **SGI** ou de **Macintosh**, les plateformes retenues doivent se distinguer sur l'un des plans énoncés plus haut. Suivant ces règles, *InterWorlds* évolue présentement dans un

environnement de type PC¹⁶, à cause du nombre de telles machines sur le marché et de leur faible coût, auquel fut ajouté des cartes d'accélération pour pallier à la lenteur du système graphique.

1.3.5.4 Les mécanismes sous-jacents

Le second élément technique qu'il importe d'adresser lors de l'ajout de fonctionnalités multiusagers à un logiciel d'immersion dans des environnements synthétiques consiste en la nature des protocoles sur lesquels les interactions s'effectueront.

Qu'il s'agisse de TCP/IP, de IPX ou de tout autre protocole, pour les mêmes raisons que pour le choix d'une plateforme d'exploitation, la rapidité, la standardisation ainsi que le taux élevé d'utilisation du protocole doivent être tenus en ligne de compte.

InterWorlds, avec son support plus que restrictif du côté du support du réseau, permet les interrogations de serveurs distants en utilisant le protocole TCP/IP. Cette particularité lui permet d'évoluer aisément sur le réseau Internet en utilisant un mécanisme portant le nom de *Sockets*¹⁷.

1.3.6 Description de l'environnement

Ce dernier élément ne peut que très directement augmenter la qualité d'un logiciel d'immersion. En effet, son apport significatif se fait plutôt sentir lors de la création des environnements synthétiques.

¹⁶ De l'anglais : *Personal Computer* ou ordinateur personnel.

¹⁷ Ce terme anglophone correspond à un mécanisme d'entrée et de sortie sur des réseaux de type TCP et UDP. Pour plus de détails sur cette technologie, le lecteur est invité à se référer à [12]

La description d'un monde virtuel, pour en faire une action accessible, doit être effectuée en respectant la structure d'un langage de définition. Ce dernier peut être très personnalisé à une application précise, permettant ainsi de faciliter la création de certains objets particuliers, ou, alternativement, le langage peut en être un standardisé, tel que le VRML, et ainsi élargir le nombre d'utilisateurs potentiels. Alors que l'adoption d'un langage normalisé augmente le degré de réutilisation d'un monde, son utilisation implique souvent de se conformer à certaines contraintes ayant trait tantôt à la complexité du langage, tantôt au manque de détails des scènes et des objets ainsi définis.

Dans *InterWorlds*, les objets de l'environnement sont simples et très souvent réutilisés. Plutôt que d'utiliser un langage de définition encombrant, il fut plutôt choisi d'opter pour un fichier de configuration simplifiant la tâche de création de l'environnement. Bien entendu, ce choix limite les types d'objets peuplant le monde tridimensionnel, mais étant donné la nature informative de l'application développée, cette limitation devenait secondaire et ce choix, justifié. Cet aspect du logiciel *InterWorlds* est approfondi à la section 3.1.9.

Dans ce premier chapitre, il fut question des aspects théoriques de la réalité virtuelle. Après avoir souligné quelques définitions du concept ainsi que les principales catégories d'utilisation de cette nouvelle technologie, le modèle générique d'un logiciel d'immersion fut présenté. Ce dernier tente de simplifier la création de telles applications en les décomposant en 5 modules distinct: **APPL**, **VWS**, **3DDB**, **3DR** et **IO**.

Maintenant que le concept de la réalité virtuelle est mieux défini, la prochain chapitre couvrira la méthodologie de la création de *InterWorlds*.

CHAPITRE 2

Méthodologie du projet

Dans ce chapitre, il sera question de tous les aspects du produit *InterWorlds*, logiciel développé dans le cadre de ce travail. Après une brève introduction, les détails des multiples facettes de ce logiciel avant-gardiste seront abordés et ce, autant au plan conceptuel que technique.

Ce chapitre se veut une référence complète sur le processus de développement entrepris lors de la conception d'*InterWorlds*. Les hypothèses, les choix et même les embûches rencontrées y seront évoqués.

2.1 Présentation schématique du système

Dans les prochaines pages, quelques schémas illustrant le fonctionnement interne d'*InterWorlds* seront présentés. Ceux-ci ont pour but de clarifier l'utilisation et la conception d'un tel logiciel d'immersion dans un environnement tridimensionnel artificiel.

2.1.1 Diagrammes

La figure 2.1 démontre de façon schématique le fonctionnement global et général du système. À l'extrême gauche de l'illustration figurent les utilisateurs du service. Chacun de ceux-ci a l'opportunité de se déplacer dans le monde virtuel à l'aide de périphériques d'immersion avant-gardistes. En fait, au stade actuel du développement de l'application, les participants peuvent choisir d'utiliser une combinaison des six types de périphériques et de capteurs de mouvements suivants :

- la souris
- la manette de jeu
- un capteur de mouvements de six degrés de liberté¹⁸
- un casque stéréoscopique
- un capteur de mouvements (pour les changements d'orientation de la tête)
- un gant numérique (comportant des capteurs pour la flexion de chacun des doigts de même qu'un capteur supplémentaire permettant de déterminer les rotations du poignet de l'utilisateur).

Bien entendu, plus le nombre de périphériques utilisé est grand, plus la sensation d'immersion dans l'environnement sera forte. Par contre, certaines contraintes physiques empêchent

¹⁸ De l'anglais : **Degrees Of Freedom**. Cette expression fait en fait référence au nombre de mouvements distincts reconnaissables par le capteur. Par exemple, une souris n'en possède que deux.

d'utiliser tous ces périphériques à la fois (il est physiquement difficile de contrôler la souris, la manette de jeu et le gant numérique simultanément!).

Tous ces périphériques sont gérés par l'ordinateur personnel, local au participant. C'est à travers le module d'entrée/sortie que cette gestion s'effectue. Une description complète de chacun de ces périphériques suivra, dans la section 2.2 décrivant les aspects techniques du projet.

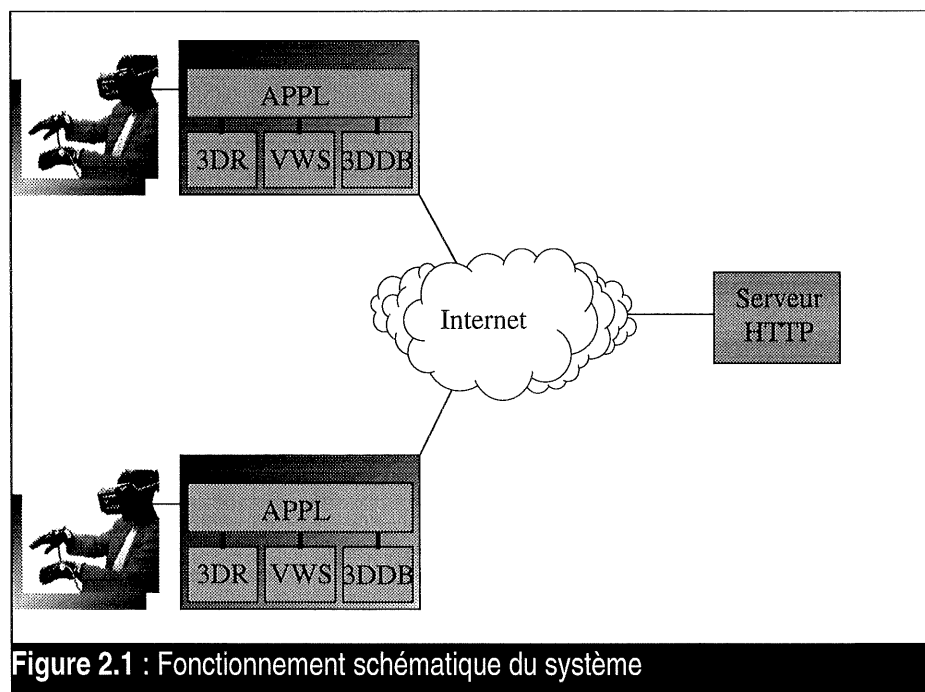
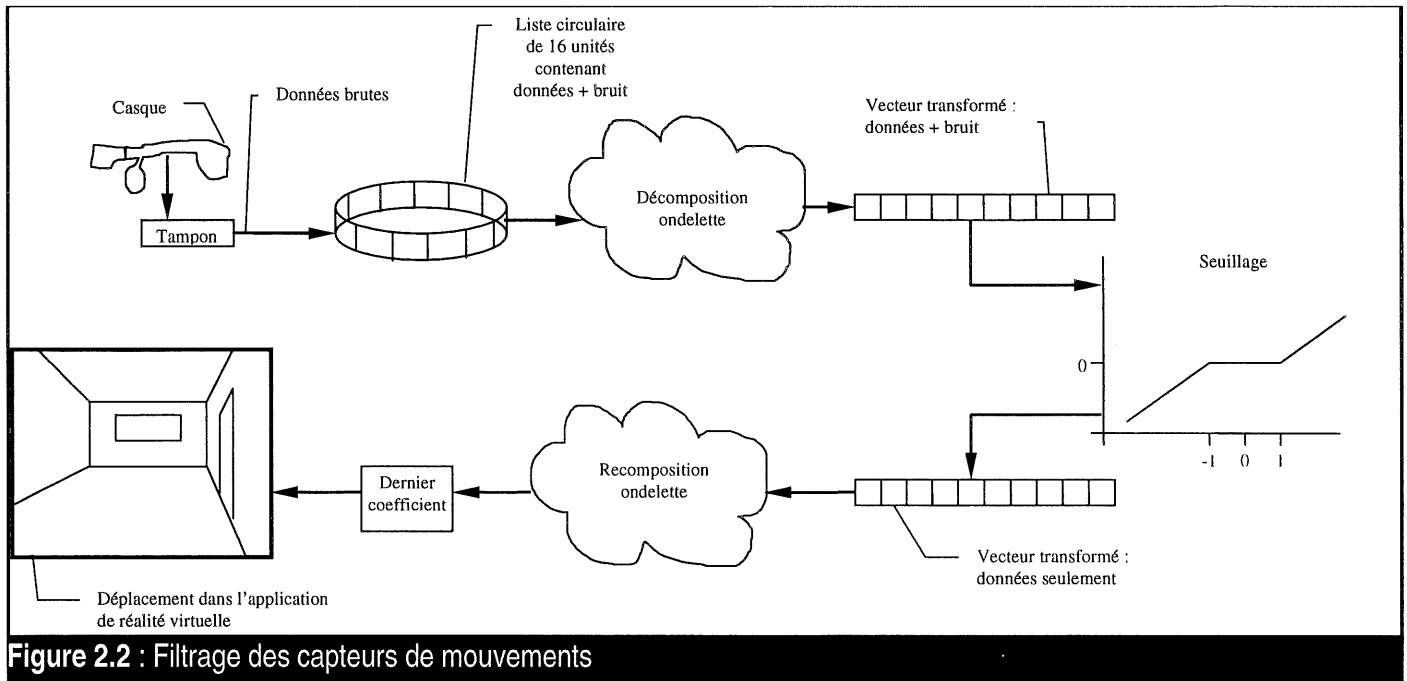


Figure 2.1 : Fonctionnement schématique du système

La figure 2.2 illustre le mécanisme logiciel intervenant lors de l'utilisation de l'un de ces capteurs de mouvement (en l'occurrence, le casque d'immersion). Périodiquement, à raison de 250 fois par seconde, les capteurs du casque font parvenir leur état à l'unité centrale de traitement hôte via le port série de l'ordinateur. Ces données, représentant des valeurs absolues de rotation autour des axes X, Y, et Z, sont relativisées, filtrées, puis fournies au module de traitement des capteurs (le module d'entrée/sortie). Ces valeurs sont ensuite interprétées et l'affichage de l'environnement est modifié en fonction de ces nouvelles informations. Dans la section 3.1, portant sur les problèmes rencontrés lors du développement

du logiciel, les phases de la relativisation et du filtrage des données des capteurs seront plus approfondies.



2.1.2 Hypothèses de base lors de la conception

Le projet *InterWorlds* est issu de l'étude de plusieurs logiciels et projets de réalité virtuelle entrepris un peu partout à travers le monde. C'est suivant cette synthèse qu'il apparut évident que certains besoins spécifiques du domaine s'avéraient toujours insatisfaits. Une ébauche du projet fut donc mise sur pied, après quoi elle fut présentée à une firme du domaine privé : Téléglobe Canada.

Téléglobe avait, à ce moment précis, un besoin imminent pour une application innovatrice permettant de démontrer l'utilisation des liens de télécommunications qu'elle venait

d'implanter entre le Canada et l'Europe. Le projet initial fut donc adapté puis soumis aux responsables de l'entreprise pour ensuite être accepté presque sans modification. Certaines contraintes matérielles et temporelles furent introduites afin de répondre aux exigences du milieu privé et c'est ainsi que le projet fut mis sur pied.

L'une des principales contraintes imposées par Téléglobe dans le cadre de ce projet fut le choix d'une plateforme d'opération. En effet, le produit final devait pouvoir évoluer sur un ordinateur de type PC afin, d'une part, de leur permettre éventuellement de réutiliser l'application et, d'autre part, de démontrer que la technologie de télécommunication qu'ils ont mise sur pied ne nécessite aucunement de machines ultra-puissantes pour s'exécuter. Il fallait opter pour des composants matériels et logiciels à la fine pointe de la technologie pour permettre une immersion convaincante malgré la "faible puissance" de la station de travail imposée. Après avoir comparé divers systèmes, le choix s'arrêta sur une machine de type TD-30 de la fabricante *Intergraph*. La station est composée de deux processeurs *Intel Pentium^{md}* cadencés à 100 Mhz, évoluant en parallèle.

Pour tirer profit de ces deux unités de traitement, *Intergraph* recommandait fortement le système d'exploitation *Windows NT^{md}*. Ce dernier était le seul environnement de travail capable de traitement parallèle sur une plateforme *Pentium* de *Intel*.

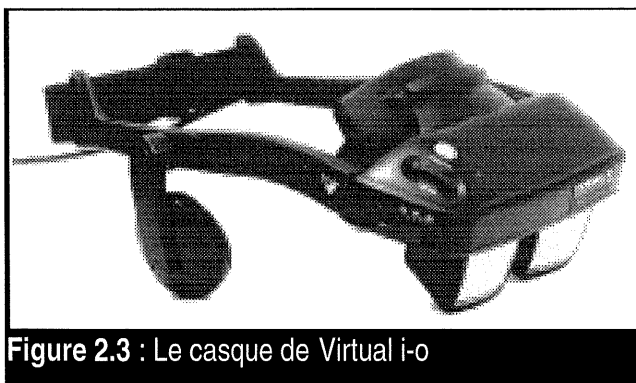
Malgré le fait que le TD-30 représentait l'ordinateur personnel le plus rapide qu'il fut possible de tester (près de deux fois plus rapide que son plus féroce rival), la station ne permettait pas de concrétiser une immersion satisfaisante, soit un rafraîchissement de l'image au minimum 15 fois la seconde pour un environnement comportant 10 000 polygones¹⁹. La solution résidait dans un accélérateur graphique compatible avec *Windows NT*. Encore une fois, c'est

¹⁹ Ces contraintes représentent le seuil minimal acceptable pour une séquence animée [13]. En deçà de quinze images par secondes, le cerveau interprète la séquence d'images comme un diaporama, plutôt que comme une séquence continue et l'effet d'immersion est grandement diminué.

Intergraph qui offrait la meilleure solution : le système intégré GLZ-5, composé d'un accélérateur géométrique jumelé à deux cartes d'accélération de textures, permet l'affichage théorique maximum de 225 000 polygones à la seconde, soit une animation composée de 15 000 polygones rafraîchis quinze fois par seconde. Le tout compatible avec la norme *OpenGL*, aussi disponible sous *Windows NT*.

L'environnement de développement intégré choisi fut celui de *Microsoft* appelé **Visual C++** : il offre une intégration complète de plusieurs outils de développement ainsi que la possibilité de créer des applications capables de tirer profit d'un environnement parallèle, si celui-ci est disponible. Bien entendu, il est aussi développé par la même firme qui a conçu *Windows NT*, le système d'exploitation utilisé dans le cadre de ce projet. Cette particularité devient en fait un avantage si l'on désire tirer parti des multiples forces du système d'exploitation pour optimiser la vitesse d'exécution du logiciel.

Du côté des périphériques, il était clair qu'un casque d'immersion ainsi qu'un gant numérique seraient nécessaires pour offrir une expérience virtuelle satisfaisante pour chacun des participants. Ainsi, après avoir comparé qualité graphique, prix et autres avantages de chaque produit, c'est le **i-Glasses** de *Virtual i-o* qui fut retenu (voir figure 2.3). C'est son faible coût, son poids léger, son aspect avant-gardiste ainsi que le fait qu'un capteur de mouvement soit intégré au casque qui en ont fait le choix idéal.



Un autre critère de sélection très important est une particularité que seul ce modèle, encore à ce jour, offre : il n'obstrue pas la vue du participant lorsque celui-ci est dans le monde virtuel. Cette qualité porte le désavantage de limiter l'effet d'immersion qui peut se faire ressentir puisque l'utilisateur, même lorsqu'il porte le casque, peut quand même voir son environnement "réel", tels le plancher et ses jambes. La résultante directe de cette limitation est d'avoir l'impression d'assister à un film interactif tridimensionnel sur un écran géant plutôt qu'à une expérience virtuelle. Par contre, c'est précisément cette limitation qui permet aux utilisateurs de naviguer dans le monde artificiel pendant de longues périodes, sans que ne surviennent des maux de cœur ou un certain étourdissement, découlant de l'immersion complète. En effet, il est reconnu qu'une immersion complète dans un environnement virtuel en utilisant un casque fermé (qui ne laisse donc pas à l'utilisateur l'opportunité de se stabiliser en se référant à un objet statique de son environnement réel) jumelé à un champs de vision plus large conduit presque inmanquablement à des étourdissements ou autres malaises, un effet direct du réflexe vestibulo-oculaire[14].

Le choix de ce modèle de casque plus conservateur fut donc très pragmatique : l'entreprise privée n'a pas intérêt à rendre malades ses clients potentiels lorsque ceux-ci veulent justement essayer les services en démonstration!

En plus de tous ces choix, une autre décision devait être prise. Il fallait alternativement développer une librairie graphique compatible avec *OpenGL* sous *Windows NT* ou acheter un ensemble de développement offrant exactement ces services. Étant donné les contraintes temporelles et les réticences des dirigeants de Téléglobe Canada à subventionner le développement complet d'une telle librairie alors que d'autres existent déjà sur le marché, c'est la seconde option qui fut retenue.

C'est le **WorldToolKit** de la compagnie *Sense8* qui fut retenu. Ce package, en plus d'être multiplateformes (c'est-à-dire qu'il existe différentes versions du logiciel, chacune permettant

de réutiliser le code source sur un ordinateur spécifique, tel *SGI*, *DEC*, *Alpha*, *PC*, etc.), supportait la norme graphique *OpenGL*, critique pour le projet. De plus, ce logiciel surclassait tous ses concurrents au niveau de la rapidité d'exécution des routines d'affichage.

Ce n'est qu'une fois ces choix arrêtés qu'il fut possible de procéder au développement du projet. Il était clair qu'il existait un besoin marqué pour de nouvelles techniques de navigation dans de massives banques de données [15][16]. Puisque la recherche textuelle devient pénible assez rapidement, c'est à partir de ce point que l'application finale prit forme. Téléglobe voulait maximiser ses investissements en matière d'implantation de câbles transcontinentaux en générant des quantités importantes de trafic sur les liens en question. Il s'avérait alors opportun de mettre sur pied un environnement virtuel permettant de consulter des séquences vidéo à distance. En ce faisant, les exigences de Téléglobe Canada étaient rencontrées (le transfert sur demande de séquences vidéo requiert une largeur de bande assez impressionnante) et une technique avant-gardiste de navigation allait être mise sur pied, redorant l'image corporative de l'entreprise.

Pour que la nouvelle méthode de navigation soit intéressante, il fallait la rendre simple à utiliser et surtout très intuitive. En créant une application de réalité virtuelle, il devenait aisé de répondre à ces deux critères. Il ne restait donc plus qu'à se concentrer sur la façon de présenter l'environnement afin de le rendre intéressant puis d'étudier diverses alternatives concernant la disposition des objets de l'environnement pour rendre la recherche intuitive. Le tout devait résulter en un logiciel interactif et autonome permettant de manipuler des objets abstraits (dans le cas présent, des séquences vidéo) afin de rendre possible une navigation et une consultation des contenus, peu importe la nature de ces derniers. Une description complète du projet figure à la section 2.2.3.

2.1.3 Travaux connexes

Hormis le fait qu'une application commerciale du genre d'*InterWorlds* n'existe pas à ce jour, il est possible de retrouver sur le marché d'autres logiciels qui accomplissent chacun une facette de ce que permet *InterWorlds*. Dans les prochains paragraphes, trois caractéristiques importantes du projet seront comparées avec d'autres logiciels déjà sur le marché.

2.1.3.1 Consultation de séquences vidéo à distance

Sans se prétendre complets d'aucune façon, les prochains paragraphes compareront certains aspects de deux logiciels de visionnage de séquences vidéo à distance. Au moment du développement du projet, seuls deux de ces produits étaient disponibles sur le marché. Le premier, offert par la compagnie *XingTech*, permet de visionner des séquences MPEG, alors que le second, VDO, nécessite d'encoder le clip vidéo dans un format spécifique, développé exclusivement pour ce produit. Alors que cette dernière technique semble permettre un meilleur taux de compression, le produit de *XingTech* utilise un encodage standardisé et permet donc de plus facilement récupérer les fichiers encodés pour une utilisation ultérieure, dans de futurs projets.

Il n'existe, sur le marché, que très peu de logiciels permettant de consulter du contenu cinématographique numérisé non-local (résidant sur un serveur distant). Depuis la création d'*InterWorlds*, quelques autres compagnies ont développé de nouvelles méthodes pour permettre une telle consultation à distance, mais l'approche de *XingTech* semble, encore aujourd'hui, des plus intéressantes étant donné son support pour la norme répandue qu'est le MPEG.

Les diverses tentatives dans ce domaine peuvent être évaluées en fonction de quelques-unes de ces caractéristiques :

- largeur de la bande passante requise,
- taille du fichier encodé et compressé,
- “effort de travail” requis pour la décompression,
- taille de la fenêtre de la séquence vidéo,
- matériel de décompression supplémentaire requis,
- temps requis lors de l’encodage,
- adaptation de la qualité du signal vidéo selon débit disponible,
- etc.

Par exemple, afin de parvenir à afficher une séquence vidéo particulière sans compression, le réseau sous-jacent doit être excessivement rapide et aussi comporter une large bande passante disponible faute de quoi le visionnage est saccadé et ressemble plutôt à un diaporama. Ainsi, le taux de compression et l’exactitude de signal vidéo produit lors de cette compression sont particulièrement critiques.

Les deux logiciels étudiés, Xing et VDO, se ressemblent certes à certains égards, mais Xing comble quelques lacunes de VDO, comme l’adaptation de la qualité de l’affichage en fonction du débit disponible. En effet, lorsque Xing doit évoluer dans un environnement où la bande passante est restreinte, la décompression des images composant la séquence vidéo est mise de côté, afin de se concentrer sur la trame sonore. Ainsi, le clip vidéo ressemble à un diaporama, mais, même dans de conditions extrêmes, le message audio peut tout de même être transmis.

Dans le cas de VDO, aucune adaptation de la qualité de la séquence vidéo ne peut être accomplie. Si la capacité du réseau ne permet pas de véhiculer la séquence, aucune décompression n'a lieu.

C'est donc la solution de Xing qui fut retenue, mais un problème persistait toujours : le logiciel est une application autonome et, afin de l'intégrer à l'intérieur du projet, il fallait obtenir un kit de développement de *XingTech* pour pouvoir ajouter les capacités de décompression à même *InterWorlds*. De cette façon, le développement parallèle d'un décompresseur MPEG devenait superflu. La section 3.1.10 traite des autres problèmes rencontrés en ce qui a trait à l'affichage des séquences vidéo.

2.1.3.2 Navigation dans les bases de données

InterWorlds, outre son aspect réalité virtuelle, n'est fondamentalement qu'un logiciel de navigation à travers de larges banques d'objets quelconques. D'autres outils logiciels effectuant un travail similaire existent actuellement sur le marché. Par exemple, les outils de consultation et de référence utilisés dans les bibliothèques et même les logiciels de gestion d'inventaire dans les grandes entreprises de vente au détail permettent de rechercher une information à travers une masse de données qui peut atteindre des proportions gigantesques.

La majorité des logiciels de consultation de larges banques de données se concentrent sur l'aspect textuel des données. Ils permettent par exemple d'effectuer des recherches en utilisant des mots-clés, de comparer de l'information contenue dans certains champs de données, etc. Ces méthodes sont tout à fait pertinentes dans certains domaines d'application, alors que, dans d'autres, l'utilisation du texte ne semble ni adéquate, ni suffisante. Aussi, dans un contexte où l'utilisateur veut consulter de l'information multimédia²⁰, un engin de

²⁰ Le terme est ici utilisé à son sens large afin d'inclure tout contenu renfermant au moins une information de nature non-textuelle (par exemple, une pièce de musique ou une séquence vidéo).

recherche ne manipulant qu'une simple description textuelle du contenu ne rend pas justice aux informations disponibles. C'est avec cette prémisse en tête que le développement d'*InterWorlds* s'est effectué. Ainsi, à l'intérieur de celui-ci, une recherche peut être active, plutôt que passive, en "plongeant" l'utilisateur dans sa recherche, à l'intérieur d'un environnement où il peut concrètement manipuler des objets représentant l'information qu'il tente de trouver. Une telle navigation permet en fait de concrétiser les concepts abstraits que l'on retrouve dans la majorité des engins de recherche textuels. Les mots sont remplacés par des images et les phrases, par des sons ou des séquences animées.

Lors de la conception d'*InterWorlds*, cette nouvelle façon de naviguer dans des banques de données était encore inconnue. Quoique quelques autres tentatives d'une telle navigation ont été réalisées depuis, il n'en reste pas moins qu'*InterWorlds* fut l'un des premiers essais dans cette direction. À en juger à l'aise que les utilisateurs ont eu à se promener dans cet environnement pour effectuer des recherches, il devient facile d'imaginer qu'un autre domaine prometteur de la réalité virtuelle est sans contredit une forme de navigation dans de larges banques de données.

2.1.3.3 Immersion dans un environnement tridimensionnel

En plus des deux aspects précédents, l'immersion dans un monde synthétique en trois dimensions est une autre facette très intéressante d'*InterWorlds*. Alors qu'il existe évidemment bien d'autres logiciels utilisant la réalité virtuelle, *InterWorlds* se distingue de par le fait qu'il insiste sur l'effet d'immersion en positionnant le participant dans l'environnement virtuel à la hauteur des yeux d'un être humain et lui permet de voir une représentation de sa main (puis éventuellement de la totalité de son corps) dans cet environnement.

Alors que d'autres tentatives comme *AlphaWorld*²¹ insistent davantage sur l'aspect multiusager d'un environnement virtuel, *InterWorlds* tente de prononcer l'effet d'immersion dans un monde alternatif en utilisant des graphiques tridimensionnels plus réalistes (en utilisant la lumière et les ombres, par exemple), des routines ne permettant pas de se déplacer à travers les murs, et plusieurs autres techniques permettant d'intensifier la sensation de se retrouver dans un environnement synthétique quasi-réel.

2.2 Description technique

Une description plus approfondie des multiples points entrevus jusqu'à présent est maintenant de mise. Dans un premier temps, les détails techniques du projet seront étalés puis, dans la section suivante, il sera question de l'espace virtuel dans lequel les participants sont plongés lors de chaque expérience d'immersion.

Le lecteur trouvera, dans les prochains paragraphes, les détails techniques des périphériques utilisés pour le projet ainsi qu'une description complète de ce dernier. De plus, un aperçu du serveur d'*InterWorlds* sera présenté.

2.2.1 Le casque d'immersion

Tel que décrit dans la section précédente, c'est le produit **i-Glasses** de la compagnie *Virtual i-o* qui fut retenu comme casque d'immersion (voir figure 2.3). Le casque lui-même comporte

²¹ *AlphaWorld* est un environnement virtuel conçu par David Gobel. Son utilisation est ouverte à tous et disponible sur le réseau Internet à l'adresse <http://www.worlds.net>.

deux entités distinctes : à l'avant, devant les yeux du participant, figure le module de sortie composé de deux affichages à cristaux liquides comportant chacun 180 000 pixels (permettant donc une résolution de 300x200 pour chacune des trois couleurs de base : le rouge, le vert et le bleu) et couvrant chacun un champ de vision équivalent à trente degrés. Le casque permet l'envoi d'un signal différent pour chaque œil en utilisant une technique dite d'entrecroisement²² où les images en entrée sont systématiquement envoyées à l'œil droit puis à l'œil gauche dans le but d'accomplir un effet stéréoscopique.

La résolution maximale acceptée par le module d'affichage est de 640x480 avec seulement seize couleurs. Pour une utilisation ludique, cette résolution peut s'avérer suffisante mais, dans le cadre d'un projet comme celui-ci, cette limitation était nettement trop contraignante. Il sera question de cette problématique dans le prochain chapitre.

À l'arrière du casque et de la tête de l'utilisateur figure la seconde moitié du casque **i-Glasses** : le capteur de mouvement. Ce périphérique permet de capter les rotations de la tête du participant autour des axes X, Y et Z. C'est un dispositif électromagnétique qui permet de reconnaître les 360 degrés de rotation autour de l'axe des Y (tourner la tête de droite à gauche et inversement) et environ 120 degrés de giration autour des deux autres axes (monter / descendre la tête et pencher la tête vers les épaules gauche et droite).

De par son design, ce casque ne permet pas une immersion totale puisqu'il laisse à l'utilisateur la possibilité de se stabiliser en conservant un point de référence à même son environnement physique immédiat. En effet, en ne couvrant pas totalement le champ de vision du participant, le casque permet à ce dernier, même lorsque l'image affichée bouge beaucoup, de regarder le plancher de la pièce dans laquelle il se trouve et ainsi prévenir l'émergence de malaises liés

²² De l'anglais : *Field Sequential*

à une désynchronisation de l'image perçue et des informations reçues par le centre de stabilisation et d'équilibre du corps humain, situé dans l'oreille interne[14].

Alors que certains considèrent que cette immersion réduite est une faiblesse du design de ce casque, d'autres persistent à croire qu'il s'agit plutôt d'un avantage de marque lorsque le casque est utilisé par des utilisateurs non-habitués, ou lors de démonstrations corporatives, sachant que les participants malades sont peut-être moins enclins à devenir des clients!

2.2.2 Le gant numérique

Un second élément quasi indispensable à une immersion adéquate est sans contredit le gant numérique. Celui-ci permet à l'ordinateur de reconnaître les mouvements manuels physiques de l'utilisateur. Certains modèles plus avancés technologiquement (et surtout plus coûteux!) comprennent plus de vingt capteurs de mouvements par main. Ainsi, avec de tels périphériques, les flexions, les extensions ainsi que les mouvements latéraux des doigts sont reconnus par les multiples capteurs et immédiatement transmis au logiciel.

D'autres modèles, tel celui utilisé dans le cadre de ce projet, ne reconnaissent qu'un sous-ensemble des mouvements saisis par les modèles haut de gamme. À ce titre, le **Fifth Glove** de la compagnie africaine *5DT* permet, à l'aide de cinq fines fibres optiques, de capter la position approximative de chacun des doigts d'une main. En effet, à travers chacun des doigts du gant passe une fibre optique parcourue d'un signal lumineux qui, même lorsque légèrement pliée, empêche le faisceau d'arriver à l'autre extrémité de la fibre. Ainsi, en mesurant l'intensité lumineuse du signal reçu et en le comparant avec la force du faisceau initialement transmis, il est possible de déterminer la "quantité de flexion" que subit un doigt. La version actuelle du gant permet de reconnaître jusqu'à 256 degrés de flexion, mais il est pertinent de noter que pour un gant fonctionnant avec ce type de mécanisme, il est impossible

de distinguer une flexion de la première articulation d'un doigt de celle de la seconde ou de la troisième articulation. C'est la quantité totale de flexion qui est rapportée, celle totalisant les flexions subies aux trois articulations de chaque doigt (ou de deux articulations, dans le cas du pouce). Les discussions des sections 3.1.1 à 3.1.5 couvrent certains des problèmes reliés au gant.

En plus des capteurs de flexions et d'extensions, le *Fifth Glove* est doté de deux autres capteurs électromagnétiques capables de reconnaître certains mouvements du poignet. De cette façon, les flexions de haut en bas et les rotations de droite à gauche de la main peuvent être transmises au logiciel permettant ainsi une immersion un peu plus convaincante en supportant un plus grand nombre de mouvements de la main du participants.

Le casque *Virtual i-o* ainsi que le *Fifth Glove* sont tous deux reliés à l'ordinateur via l'interface offerte par les ports de communication sériels de la machine.

Maintenant que ces deux périphériques d'immersion sont mieux définis, il paraît approprié, à ce point, de décrire les aspects techniques de la version cliente du logiciel *InterWorlds* et d'explicitier chacun de ces divers modules.

2.2.3 *InterWorlds* Client

InterWorlds est un ensemble logiciel permettant à des utilisateurs reliés sur Internet de s'immerger dans un environnement tridimensionnel virtuel. Cette solution intégrée est composée de deux modules principaux : le client et le serveur. C'est, bien entendu, le serveur qui distribue les mondes virtuels aux stations clientes, mais avant de le décrire plus en détail, il serait opportun de mieux décrire le module d'interaction : celui de la station cliente.

L'interaction de chaque participant se fait grâce au module client. Ce module peut être décomposé en quatre parties distinctes : le bloc de gestion des périphériques d'entrée/sortie, le bloc d'affichage tridimensionnel, l'entité d'interface avec le réseau et la base de données d'objets géométriques. Ces entités sont gérées par un module supplémentaire les chapeautant toutes : l'application. C'est donc cette mosaïque logique qui constitue le module client (voir figure 2.1).

Ces modules peuvent être apparentés avec les notions théoriques abordées à la section 1.2 traitant de la composition d'une simulation. En effet, l'application correspond au module **User Application**, le bloc de gestion des périphériques d'entrée/sortie correspond au **I/O Devices**, le bloc d'affichage tridimensionnel au **3D Rendering** et la base de données d'objets géométriques au **3D Geometric Database**. Pour ce qui est de l'entité d'interface avec le réseau, ses fonctionnalités sont, quant à elles, retrouvées à même le module **User Application**.

Il est à noter qu'étant donné la nature statique des éléments de la simulation, l'entité théorique **Virtual World Simulator** ne se retrouve pas dans *InterWorlds*. Bien entendu, les objets sur lesquels le participant peut "cliquer" possèdent des propriétés particulières qui se retrouvent dans la base de données d'objets géométriques. Par contre, le **VWS** n'est aucunement requis puisque c'est plutôt le module d'application qui se charge de déclencher les actions appropriées en fonction des sélections de l'utilisateur.

Alors qu'une analyse de la conception du logiciel sera couverte dans le chapitre suivant, une description du contenu de l'environnement créé par *InterWorlds* client figure dans la section 2.3.

2.2.4 InterWorlds serveur

Lors de la conception originelle du système, la seconde moitié du logiciel, le serveur, permettait à plusieurs participants d'utiliser et d'interagir avec un environnement virtuel et de communiquer avec les autres utilisateurs. Par contre, étant donné les contraintes temporelles de livraison imposées par l'entreprise privée, l'idée fut rancardée et c'est avec des fonctionnalités réduites que le serveur a dû prendre naissance. En fait, ses fonctions étaient devenues tellement minimales qu'il n'agissait plus que comme serveur HTTP²³, acceptant des requêtes et y répondant par la transmission de fichiers de description du monde²⁴. Plutôt que de réinventer la roue, c'est précisément ce qu'advint au module serveur : il fut abandonné et ses fonctions sont présentement prises en charge par un serveur HTTP typique.

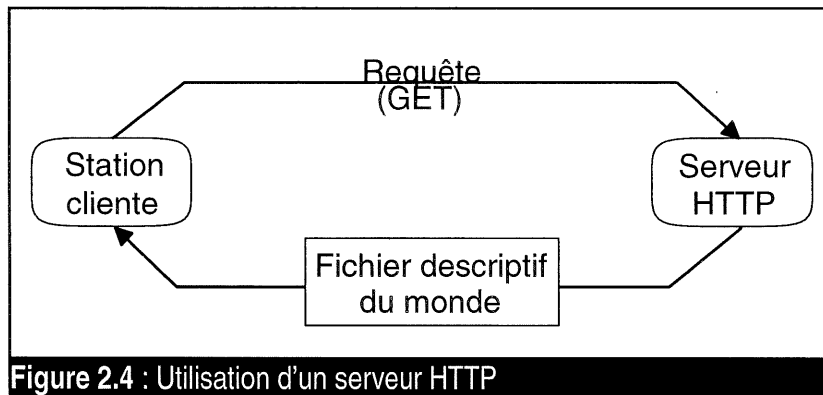


Figure 2.4 : Utilisation d'un serveur HTTP

La figure 2.4 illustre le processus impliqué lors de chaque demande de téléchargement d'un fichier descriptif du monde virtuel par une station cliente. Lorsque cette dernière effectue une requête, à l'aide de la méthode **GET** du protocole HTTP, le serveur retourne le fichier exigé en guise de réponse.

²³ Le *Hyper-Text Transport Protocol* est le protocole utilisé pour transférer des pages HTML sur le WWW qu'est Internet.

²⁴ Il sera question de ceux-ci dans la section 3.1, portant sur les problèmes rencontrés lors de l'achèvement du projet.

Il n'est cependant pas dit que l'idée première du serveur ne fera pas surface éventuellement. Tel qu'il fut soulevé dans le chapitre précédent, l'évolution logique du concept de la réalité virtuelle se situe dans la possibilité d'interagir avec d'autres participants à même le monde tridimensionnel virtuel. L'ajout de capacités de fonctionnement en réseau demeure donc très pertinent et c'est suivant cette idée maîtresse que la future refonte du serveur s'effectuera.

2.3 Description de l'espace virtuel

Dans la prochaine section, il sera question de l'environnement synthétique dans lequel les utilisateurs d'*InterWorlds* se déplacent et consultent les contenus de la banque de données. Une séance d'immersion typique y sera aussi décrite.

Tel que précédemment décrit, *InterWorlds* plonge l'utilisateur dans un monde tridimensionnel représentant en quelque sorte un musée. L'intérieur de celui-ci est divisé en pièces distinctes, chacune correspondant à un thème particulier (dessins animés, clip vidéo corporatifs, faune sauvage, etc.). La figure 2.5 illustre partiellement ces divisions. Elle rapporte ce que le participant voit lorsqu'il se retrouve dans le hall d'entrée de l'environnement : plusieurs pièces, chacune identifiée par une image descriptive correspondant à son contenu, communiquant à une chambre principale.

En effet, chacune de ces pièces, nommées "ailes", est caractérisée par un thème, illustré par une image située en haut, à gauche de la porte d'entrée de l'aile. Pour l'aile de la faune sauvage, par exemple, c'est l'image d'un bébé béluga qui laisse deviner le contenu de la pièce. Parfois, lorsque les images ne parviennent que trop mal à décrire le contenu d'une salle, il devient intéressant d'utiliser un médium descriptif supplémentaire : l'audio. *InterWorlds*

permet effectivement d'ajouter une description auditive du contenu des pièces. Cette description est disponible lorsque le participant s'approche d'un de ces panneaux de description et le sélectionne avec le gant numérique. Automatiquement, la bande sonore est jouée, permettant ainsi de préciser le contenu de l'aile en question.

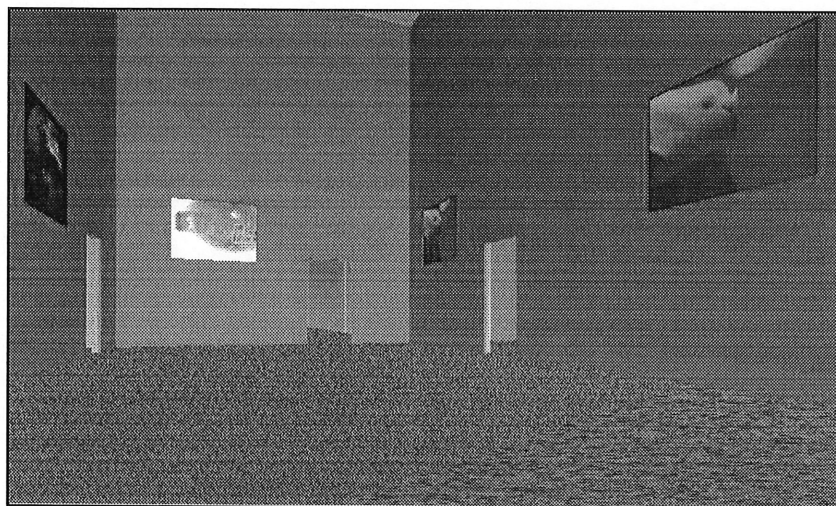


Figure 2.5 : L'intérieur de l'environnement

La navigation dans *InterWorlds* se fait en utilisant le concept du “*First-person view*”. Cette notion permet au participant de voir l'environnement virtuel et de s'y promener comme s'il s'y retrouvait physiquement. Pour cette raison, il est impossible de voir le “corps virtuel” complet de l'utilisateur dans le monde synthétique, puisque le point de vue (la caméra) est situé dans les yeux de ce personnage. Ce type d'expérience est beaucoup plus immersif que si la navigation s'effectuait avec un autre type de vue (à vol d'oiseau, vue $\frac{3}{4}$, etc.) [17].

La sélection des objets dans l'environnement d'*InterWorlds* se fait, idéalement, en utilisant le gant numérique, mais peut aussi se faire avec une souris, si un tel gant n'est pas disponible. Alors que le casque d'immersion sert exclusivement lors de l'*orientation* dans l'environnement, le gant est effectivement le périphérique principal de *navigation* dans ce monde virtuel. Il permet donc de se déplacer, mais aussi de sélectionner les divers objets

constituant l'espace artificiel. Les multiples facettes de l'utilisation de ce gant à l'intérieur d'InterWorlds seront décrites plus en profondeur dans le chapitre suivant.

Selon son désir, le participant pourra alors se diriger vers l'aile de son choix, en utilisant le gant comme outil de navigation, pour aboutir sur une pièce ressemblant à celle illustrée à la figure 2.6. Dans cette dernière, le participant a la liberté de choisir parmi quatre images, représentant chacune une information de la banque de données. À chacune d'elle est donc associée une séquence vidéo, et, pour démarrer le visionnage, il suffit à l'utilisateur de sélectionner l'image avec le gant. Aussitôt la sélection effectuée, *InterWorlds* envoie une requête au **serveur vidéo**²⁵ l'informant de la sélection de l'utilisateur. Sur réception, un outil de visionnage MPEG est démarré et permet ainsi au participant de démarrer la consultation de la séquence vidéo de son choix.

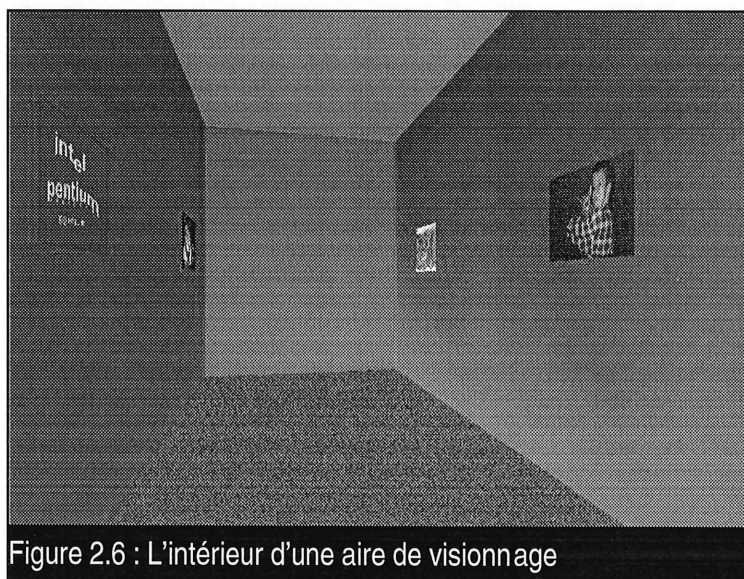


Figure 2.6 : L'intérieur d'une aire de visionnage

Tout au long de la consultation, l'utilisateur peut contrôler la vitesse du visionnage (avancement rapide, retour arrière, interruption, etc.) et même continuer à se déplacer dans

²⁵ Le serveur vidéo est un autre projet de Téléglobe. Il consiste en un ordinateur équipé pour diffuser du contenu vidéo MPEG sur Internet, au moyen de liens de communication à haut débit.

l'environnement, toujours en utilisant le gant et le casque d'immersion. Il peut aussi sélectionner une autre image et déclencher le démarrage d'une autre séquence vidéo, qui remplacera celle s'affichant présentement.

Le processus de création automatique des ailes du musée se fait en utilisant un gabarit prédéterminé. Un fichier contient en effet un mur et un plancher type qui serviront de base pour toutes les entités similaires créées dans l'environnement. Il serait donc relativement simple d'ajouter des textures ou de modifier les couleurs des pièces.

À la fin de la simulation, soit après une période de temps prédéterminée dans le *Registry*²⁶, *InterWorlds* coupe le contrôle de la navigation et éloigne progressivement l'utilisateur de l'environnement pour lui faire comprendre que la simulation est terminée et qu'il doit laisser la place à un autre participant. Plutôt que de faire un arrêt brusque et immédiat, il fut remarqué qu'un recul progressif était moins "frustrant" pour l'utilisateur et le poussait à retirer lui-même son casque et son gant pour mettre fin à son expérience. En utilisant cette technique, les opérateurs n'ont pas à exiger du participant qu'il termine sa session, puisque ce dernier le fait de lui-même. Pendant les quelques dix secondes que dure ce recul forcé, l'utilisateur peut continuer à regarder autour de lui, mais ne peut ni avancer, ni sélectionner les objets. Graduellement, il voit le musée à vol d'oiseau, puis, après quelques secondes, le voit disparaître dans l'obscurité.

²⁶ Voir section 3.1.8 pour plus de détails à ce sujet.

CHAPITRE 3

Conception du logiciel

Ce chapitre se veut une continuation du second en ce sens qu'il utilise les concepts techniques décrits dans le chapitre précédent pour expliquer certaines situations rencontrées lors du développement. En effet, avant de dresser une liste sommaire des multiples améliorations dont bénéficierait *InterWorlds*, une énumération des problèmes rencontrés de même que des solutions retenues et des alternatives étudiées sera divulguée.

3.1 Problèmes, difficultés et solutions

Dans les prochains paragraphes, il sera question des problèmes qui sont survenus tout au long de la conception et du développement du logiciel. Chaque problème sera décrit et la situation problématique de chacun sera expliquée. De plus, une liste des alternatives envisagées, de même que la solution retenue, sera énoncée.

3.1.1 Sensibilité des capteurs de mouvements

Étant donné la nature électrostatique des capteurs de mouvements, il est fréquent de remarquer du *bruit* dans le signal que ceux-ci retournent. Ces fluctuations indésirables se traduisent par des mouvements saccadés de l'image transmise à l'utilisateur. Ainsi, alors que sa tête est véritablement stable et fixe, l'information visuelle que le participant reçoit du casque lui suggère qu'il en est autrement. Selon l'image reçue, l'utilisateur a plutôt l'impression que sa tête bouge par brefs déplacements chaotiques alors qu'il n'en est rien. Cette disparité entre la situation réelle et la simulation est, en plus d'être excessivement frustrante, l'une des principales cause des fameux maux d'immersion prolongée, tel qu'il fut question dans la section précédente.

Alors que la diminution de la sensibilité des capteurs de mouvements semble une solution appropriée à ce problème, la réalité est tout à fait autre. Une telle diminution rend automatiquement impossible les légers mouvements précis de l'utilisateur, prononçant du même coup un écart entre le comportement du participant et les stimuli visuels qui lui sont retournés (avec une sensibilité réduite, un déplacement de la tête n'entraîne aucune modification de l'image affichée, désorientant d'autant plus le virtuanaute²⁷).

Si la solution ne réside pas dans la sensibilité du capteur, c'est donc dans le traitement des données que celui-ci fournit qu'une étape doit être ajoutée. À cet effet, les notions de base en traitement de signal indiquent qu'il suffit d'utiliser le filtrage ou la transformation du signal original (ou une combinaison de ces techniques) pour retirer le bruit du signal entrant [18]. De cette façon, les éléments non-désirés du signal sont simplement ignorés et les mouvements précis de faible amplitude sont tout de même conservés.

²⁷ Les *virtuanautes* sont des cybernautes virtuels qui en fait sont des cyber-astronautes virtuels!

Précisément à cette fin, plusieurs méthodes furent donc essayées. À ce titre, deux méthodes de filtrage en particulier furent envisagées : la première est celle du retranchement systématique de tous les pics et pointes du signal du capteur, alors que la seconde en est une d'interpolation des valeurs. Finalement, suivant les insuccès des tentatives précédentes, une dernière méthode fut implantée. Celle-ci consiste en une transformation pure et simple du signal en utilisant une technique appelée “ondelettes”²⁸.

Avant d'entrer dans les détails de chacune de ces méthodes, certaines particularités du capteur de mouvement intégré au casque d'immersion de *Virtual i-o* doivent être soulevées. Tout d'abord, afin de faciliter le positionnement initial du point de vue du participant dans l'environnement tridimensionnel virtuel, les données absolues du positionnement du casque devaient être relativisées. Ainsi, un mouvement de la tête de la gauche vers la droite de cinq degrés devait être représenté par la valeur “+5” alors qu'un déplacement dans la direction inverse le serait par une valeur complémentaire, soit “-5”. Cette logique devrait aussi se retrouver pour les deux autres axes de rotation : soit une valeur positive pour “monter” la tête (regarder vers le haut) et de même pour un roulis de la gauche vers la droite. De façon similaire, “descendre” la tête, ainsi qu'un roulis de la droite vers la gauche, devrait correspondre à une valeur négative.

Déjà à ce point, un léger obstacle devait être surmonté : puisque le casque permet des déplacements complets de la tête de gauche à droite, soit plus ou moins 180 degrés avec comme point “mort” (le zéro absolu) le Pôle Nord magnétique, il existe un point où l'utilisateur peut effectuer une rotation d'un seul degré vers la droite et le capteur retournera les valeurs -179 puis +179. En effet, si le participant tourne le dos au Nord magnétique les valeurs reçues du casque doivent être ajustées si elles sont relativisées, ceci afin d'éviter de sérieux problèmes de désorientation.

²⁸Dans la littérature, le terme anglais *wavelets* est davantage utilisé.

Ce problème ne se pose pas pour les deux autres axes de rotation puisque le capteur du casque ne reconnaît que 120 degrés (soit plus ou moins 60 degrés autour du point central) autour de ces axes. Cette limitation du matériel ne devient évidente que si l'utilisateur décide, par exemple, de se promener sur la tête, les pieds dans les airs!

La relativisation des données s'est faite de façon tout à fait normale, soit en conservant la dernière valeur obtenue après chaque itération et en la comparant avec la prochaine. Dans *InterWorlds*, une fonction autonome effectue précisément ce travail : `WTsensor_relativizerecord`. L'utilisation de cette dernière est reproduite à l'annexe C. Il est à noter qu'un test supplémentaire devait être effectué afin de vérifier le cas d'exception, soit lorsque l'utilisateur fait face au Sud, ceci afin de gérer la transition entre les extrêmes.

Tout en utilisant cette relativisation, la première technique de filtrage consiste simplement à ignorer les valeurs relatives supérieures à quinze²⁹ degrés. Cette technique tire son fondement du fait que le bruit électromagnétique introduit du bruit absolu plutôt que relatif dans le signal, provoquant ainsi des mouvements relatifs élevés. La faiblesse évidente de cette méthode réside dans le fait que les vives rotations de plus de quinze degrés sont filtrées et donc simplement ignorées par le module d'entrée/sortie.

La seconde technique, un peu plus efficace celle-là, filtre le signal en interpolant des données réelles de celui-ci. Par exemple, si le signal en entrée correspondait à la séquence {+4, +10, +2 et -2}, le signal filtré serait devenu {+7, +6, 0}, soit $(+4 + +10)/2 = +7$, $(+10 + +2)/2 = +6$, $(+2 + -2)/2 = 0$. Des tests furent effectués avec des rapports plus draconiens (division par trois, quatre, cinq et plus), mais le résultat obtenu était loin de convenir à la situation puisqu'il

²⁹ C'est la valeur arbitraire qui fut choisie. Elle semble à la fois correspondre à une quantité de mouvements anormalement élevée (mouvement très saccadé) ainsi qu'à un certain minimum en ce qui concerne la quantité de bruit électromagnétique introduit.

avait pour effet de diminuer l'intensité d'un mouvement, nuisant par le fait même à l'instantanéité des réponses à l'intérieur d'*InterWorlds*.

Après avoir testé ces quelques méthodes de filtrage sans avoir obtenu de succès significatif, l'idée d'adapter le filtrage par ondelettes à un signal en temps réel devenait de plus en plus intéressante. Le chapitre suivant relate comment un tel type de filtrage s'est effectué tout en expliquant le cadre théorique régissant cette technique novatrice.

3.1.2 Ergonomie des capteurs

Un autre point faible du logiciel réside dans le caractère peu intuitif du déplacement dans le monde artificiel. De façon similaire à la maladresse éprouvée par une personne manipulant un curseur à l'aide d'une souris pour la première fois, les virtuanauts d'*InterWorlds* sont confrontés à une navigation dans un environnement tridimensionnel où les seules méthodes de contrôle sont les doigts, le poignet et la tête.

Une fois les capteurs de mouvements bien pris en charge par le module d'entrée/sortie et que ceux-ci ont été stabilisés à l'aide de filtres dédiés (tel que décrit dans la section précédente), les senseurs viennent ajouter un nouveau potentiel à l'application : celui de pouvoir être ergonomique et intuitive à utiliser. En effet, les capteurs ne permettent rien de plus que la transmission des mouvements d'une certaine partie du corps du participant au logiciel d'immersion afin de lui permettre un certain degré d'interaction avec son environnement synthétique. Au-delà de ce transfert d'informations, il est primordial de donner une signification à ces données, précisément afin de rendre le logiciel plus "naturel", plus intuitif.

Pour ce faire, il est requis de déterminer un nombre fini d'états dans lequel un capteur peut se trouver, ensuite de quoi il faut assigner à chacun de ces états une action particulière à effectuer

dans l'environnement. Par exemple, lorsque les informations reçues des capteurs de mouvements du gant numérique indiquent que tous les doigts de l'utilisateur sont pliés à l'exception de l'index, il pourrait être approprié de faire "avancer" le personnage dans son monde tridimensionnel virtuel.

Une situation problématique surgit à ce point : quelles actions doivent être associées avec quels états des senseurs de mouvements? Il fallait, dans un premier temps, déterminer une façon de se représenter les différents états de façon globale, sans avoir à utiliser, par exemple, les valeurs absolues des capteurs à chaque référence. En annexe D figure une partie du code qui permet justement d'abstraire cette information des multiples états des capteurs de mouvements du gant numérique.

Le tableau 3.1 dresse une liste des combinaisons des différents états possibles du gant numérique. La première colonne de la table relève les différentes positions qu'il est possible de réaliser en utilisant les capteurs situés dans les cinq doigts du gant numérique. Il existe donc $2^5 = 32$ combinaisons différentes possibles avec les doigts.

Tableau 3.1 : Énumération des états possibles du gant	
Position des doigts (mouvements exclusifs)	États des capteurs du poignet
Pointer un seul doigt [5 façons]	Rotation gauche-droite [3 façons] Flexion / extension [3 façons]
Pointer avec deux doigts [10 façons]	
Pointer avec trois doigts [10 façons]	
Pointer avec quatre doigts [5 façons]	
Pointer avec cinq doigts (main plate) [1 façon]	
Pointer avec aucun doigt (poing) [1 façon]	
Total : 32 états disponibles	Total : 9 états
Nombre total de positionnements possibles avec le gant : 288 ($9 * 32$)	

En plus des capteurs situés dans les doigts, le gant numérique possède aussi des capteurs permettant de reconnaître les mouvements du poignet du participant. La colonne de droite du

tableau 3.1 décompose les neuf combinaisons de mouvements du poignet qu'il est possible de distinguer. Alors que le premier capteur reconnaît les rotations de la gauche vers la droite et vice versa, le second permet de capter les flexions et les extensions du gant. Ainsi, avec trois états différents pour chacun des deux capteurs, il est donc possible de reconnaître $3^2 = 9$ états distincts.

En combinant les états des capteurs de mouvement du poignet et ceux des doigts, il est donc possible de distinguer près de trois cents ($2^5 \times 3^2 = 288$) états différents. Bien entendu, la majorité de ces états sont, au mieux, peu intuitifs et souvent même difficiles à exécuter. C'est pourquoi, de cette longue liste, seuls quelques-uns ont été retenus. Le tableau 3.2 dresse la liste des gestes retenus et précise aussi l'action associée à chacun de ceux-ci.

Tableau 3.2 : Association des gestes du gant et des actions	
États (gestes)	Déplacements
Pointer avec un doigt	Avancer
Pointer avec deux doigts	Reculer
Poing ou main ouverte	Arrêter
Fusil	Sélectionner
Pouce vers le haut	Monter
Pouce à droite ou à gauche	Déplacement latéral

Cette association des gestes et des déplacements sembla la plus populaire (et la plus intuitive), suite aux multiples essais effectués par des participants à l'intérieur d'*InterWorlds*.

Il est pertinent de noter, à ce point, que suite à l'implantation à l'intérieur d'*InterWorlds*, il fut remarqué que pour la majorité des gestes obtenus par le positionnement des doigts, l'association supplémentaire avec un état des capteurs de mouvements du poignet rendait le geste trop complexe et par conséquent peu intuitif et surtout mal utilisé. Ainsi, si le participant pointe avec son index, que son poignet soit fléchi, en extension, qu'il soit en

rotation vers la droite ou la gauche n'influe en rien l'action associée, qui est celle d'avancer. Pour cette raison, les capteurs impliqués dans la reconnaissance de l'état du poignet sont, dans *InterWorlds*, inutilisés.

Il pourrait par contre être intéressant de créer de nouveaux périphériques d'entrée/sortie capables de mieux reconnaître ces mouvements et ainsi permettre des déplacements plus intuitifs à l'intérieur de l'environnement synthétique.

Dans les présentations publiques où plusieurs participants se succèdent à une cadence élevée, il était pénible de paramétrer le système pour chacun des nouveaux participants. C'est pourquoi, dans de telles situations, la configuration du tableau précédent était utilisée. Par contre, suite à cette imposition de contraintes, les visiteurs se déplaçaient de façon saccadée et requéraient constamment une assistance humaine externe pour se déplacer dans l'environnement. La situation idéale serait donc de simplement demander à chaque utilisateur d'accomplir le mouvement qu'il préfère pour effectuer chacun des déplacements reconnus en utilisant l'une des techniques de la section 3.1.3. De cette façon, la navigation deviendrait intuitive pour chacun, et l'utilisateur pourrait ainsi se concentrer sur l'environnement tridimensionnel dans lequel il se déplace, plutôt que sur la façon de s'y déplacer.

Dans une prochaine version, il pourrait être intéressant d'ajouter des états *transitionnels* à la liste de ceux déjà mentionnés. Ceux-ci devraient inclure des mouvements (plutôt que de strictement permettre des états prédéterminés, tel que le fait actuellement *InterWorlds*), comme par exemple la transition entre la main complètement ouverte et le poing, pour représenter l'action de sélectionner un objet dans le monde artificiel. De tels ajouts auraient pour effet certain d'augmenter l'aspect intuitif des actions dans la navigation à l'intérieur d'un tel environnement virtuel.

3.1.3 Calibrage des capteurs

Hormis l'ergonomie des capteurs de mouvements, le calibrage de ceux-ci posa aussi quelques problèmes. Le terme calibrage correspond ici à un ajustement personnalisé de la reconnaissance de l'amplitude des mouvements pour chaque utilisateur. Étant donné les multiples différences physiologiques entre chaque individu, si les capteurs sont configurés pour reconnaître les mouvements d'un seul utilisateur ou même ceux de la majorité des participants, la quasi-totalité des autres individus utilisant *InterWorlds* serait pratiquement incapable de se déplacer dans l'environnement tridimensionnel; ceux-ci n'étant physiquement pas en mesure d'effectuer les mouvements requis ou dans l'amplitude requise pour déclencher les actions associées aux divers états des capteurs. Pour mieux saisir ce concept, il suffit de se rendre compte que la quantité de flexion maximale peut varier d'individu en individu, puisque étant reliée à certaines caractéristiques physiologiques des doigts de ceux-ci.

Ainsi, pour permettre une navigation plus naturelle pour chacun, les données brutes des capteurs de chaque participant ont dû être conservées pour chacun des états retenus (main ouverte, main fermée, index pointé, etc.). Pour ce faire, les valeurs minimales et maximales de chaque capteur furent récoltées pour chacun des utilisateurs. Cette cueillette est effectuée de façon interactive, soit en demandant à chaque utilisateur, avant même de débiter la navigation, d'effectuer chacun des gestes retenus³⁰ pendant quelques secondes. Étant donné le faible nombre d'états retenus, le calibrage n'était pas perçu comme une nuisance, mais bien comme une partie intégrante de "l'expérience virtuelle". En effet, les participants semblaient bien y prendre plaisir, et cette courte période de configuration leur permettait de mieux se familiariser avec les capteurs.

³⁰

Les gestes permettant d'obtenir l'état particulier des capteurs, voir tableau 3.2.

Deux méthodes de calibrage ont été utilisées, mais avant de décrire chacune de ces techniques, il semble approprié d'explicitier le processus de calibrage. Chaque calibrage est constitué de la cueillette des données brutes de certains capteurs. En effet, afin de calibrer le gant pour reconnaître le mouvement du poing, il faut obtenir les minima et les maxima des capteurs concernés (ceux des doigts, dans ce cas-ci). Ceci est nécessaire afin de déterminer la plus grande et la plus faible amplitude de flexion des doigts correspondant à ce geste, de l'utilisateur actuel (une valeur se situant dans l'intervalle [0..255]).

Même en sachant qu'une flexion complète des capteurs des doigts du gant correspond à la valeur brute 255, les différences physiologiques des mains des participants font en sorte que, pour certains, une flexion maximale ne correspond qu'à la valeur 200, par exemple. De plus, à certaines occasions, ce même utilisateur désirant effectuer le même geste ne parviendra pas à fléchir ses doigts dans la même mesure, et les capteurs ne retourneront alors que 150, par exemple. Ces deux nombres correspondent, respectivement, au maximum et au minimum pour ce geste, pour cet utilisateur.

La première technique de calibrage consiste à obtenir tous les maxima de tous les mouvements puis, une fois cette étape terminée, cueillir les minima pour les mêmes états. De cette façon, en exigeant du participant qu'il effectue à deux reprises les mêmes gestes, cela permet au système de mieux reconnaître les différentes amplitudes que le participant peut accomplir pour effectuer un geste particulier. En effet, avec un échantillonnage plus grand, les mouvements reconnus correspondent mieux aux gestes réels du participant.

La seconde méthode consiste à demander à chaque participant de faire chacun des mouvements pendant quelques secondes, d'effectuer une cueillette des résultats, puis d'interpréter les minima et les maxima à partir des données obtenues pendant cet intervalle. En procédant ainsi, le système peut utiliser plusieurs valeurs minimales et en tirer certaines conclusions, ensuite de quoi il procède de façon similaire pour tous les maxima.

La technique retenue dans *InterWorlds* est la seconde, celle exigeant du système de retenir le minimum parmi les minima recueillis et le maximum des maxima obtenus. L'utilisateur n'est donc appelé à configurer les gestes qu'une seule fois, ce qui semble le plus approprié pour une utilisation dans un contexte d'exposition publique, tel qu'il fut le cas pour ce projet. Pour procéder au calibrage sans exiger d'intervention externe majeure, une touche du clavier de la station cliente fut réservée à l'appel de la routine de calibrage. Ainsi, à chaque nouveau participant, il suffisait d'appuyer sur ladite touche pour démarrer le processus de configuration des capteurs. Quoique appropriée pour les salons et les expositions publiques, cette méthode n'est cependant pas idéale quant au degré de reconnaissance des mouvements effectués. En effet, la première méthode est beaucoup plus efficace, mais ne semble pas plaire outre mesure aux virtuanauts occasionnels, ceux qui n'utilisent le système que pour quelques minutes.

Dans une version ultérieure d'*InterWorlds*, il pourrait être intéressant d'opter pour une tierce méthode de calibrage : la mise à jour dynamique. Cette dernière consiste à continuellement mettre à jour les valeurs des minima et des maxima tout au long de la navigation du participant. Ceci pourrait simplement signifier d'effectuer un traitement continu sur les données de chacun des capteurs. Pendant ce traitement, le minimum de l'amplitude d'un mouvement serait réévalué en fonction des données brutes reçues du capteur. Le même ajustement serait effectué pour le maximum de chacun des mouvements, permettant ainsi de mettre à jour les amplitudes des gestes reconnus. De cette façon, la fatigue, ou tout autre élément externe susceptible de modifier la quantité de flexion des doigts, ne pourrait plus nuire à la reconnaissance des gestes, et du même coup, à la navigation dans l'environnement. Cette méthode permettrait d'éliminer la période de calibrage et tiendrait en ligne de compte un certain degré de fatigue que peuvent éprouver les participants après quelques minutes de navigation. Par contre, il serait à prévoir que l'exactitude et la précision des gestes reconnus pourraient en souffrir.

3.1.4 Flexion des articulations

Le prochain problème relève plutôt de l'aspect matériel de l'équipement d'entrée/sortie utilisé. En effet, le gant numérique Fifth Glove de *5DT* possède, tel que décrit un peu plus tôt dans ce chapitre, cinq fibres optiques miniatures permettant de capter les flexions et les extensions des doigts. Les deux extrémités de ces fibres sont reliées à un module électronique situé sur le gant, à la hauteur du poignet. Chacune des cinq fibres est dirigée sur toute la longueur d'un des doigts à l'aide de guides fixés à même le gant. Cette particularité permet à la fabricante de pouvoir offrir un tel gant à prix abordable, mais, du coup, introduit une limitation matérielle au niveau du gant : il devient impossible de distinguer la flexion d'une seule articulation avec la flexion de plus d'une d'entre elles.

Cette limitation est attribuable au fait que le capteur de mouvement, constitué d'une simple fibre optique, ne fait que retourner le rapport entre la quantité de lumière émise sur la fibre et celle reçue, après avoir parcouru la fibre. Si cette dernière n'est pas pliée par une flexion de l'une des articulations du doigt, la quantité de lumière reçue sera la même que la quantité émise³¹. Par contre, à la moindre flexion du doigt, le faisceau lumineux perd de son intensité (suite à la friction avec les rebords de la fibre) et cette fluctuation est rapportée au système par le capteur. Étant donné qu'avec cette technique il est impossible de déterminer l'endroit exact de la flexion, la distinction entre une flexion simple et une autre impliquant plus d'une articulation est tout aussi impensable.

Pour pallier à cette limitation, il fallait que l'image de chaque doigt, respectivement dirigée par un capteur sur le gant, effectue une flexion complète graduelle de toutes les articulations pour introduire un minimum de disparité entre l'image de l'écran et la situation réelle des doigts du

³¹ Un certain calibrage matériel permet d'ajuster la valeur retournée afin qu'elle ne soit pas influencée par la perte d'intensité lumineuse produite au bout de chaque doigt, alors que la fibre est pliée à 180 degrés afin qu'elle puisse revenir vers le poignet.

participant. Ainsi, la moindre flexion de l'une des articulations d'un doigt se traduit visuellement par une flexion de toutes les sous-parties d'un doigt : les phalanges. Pour parvenir à effectuer cette flexion multiple, il fallait toutefois décomposer chaque doigt en trois segments (dans le cas de l'index, du majeur, de l'annulaire et de l'auriculaire) ou en deux segments pour le pouce. Chaque segment doit aussi être en mesure d'effectuer une rotation dans un axe lui conférant un certain aspect naturel. Pour les quatre doigts, le point de rotation des segments correspond au milieu et au bas de la face **supérieure** de chaque segment, ceci afin de ne pas introduire des "trous vides" lors des flexions. En effet, si le point de rotation de chaque segment est situé au centre géométrique de la phalange, la rotation s'effectue autour d'un point non-commun avec la phalange inférieure et introduit donc des vides entre celles-ci.

En plus de cet ajustement du point de rotation, il est aussi important de s'assurer qu'une trop grande flexion d'un doigt, au delà du seuil maximal fixé lors de l'étape du calibrage³², ne soit pas représenté dans l'environnement synthétique par des doigts qui effectuent une rotation trop prononcée, allant jusqu'à passer outre le niveau de la paume et donnant l'impression de s'enfoncer à l'intérieur de la main. En effet, pour pallier à cette situation, il s'avéra important de fixer un seuil maximum de rotation pour chaque segment de doigt. Ce seuil, fixé à 90 degrés, aura donc pour effet de limiter la rotation des phalanges virtuelles autour de la paume de la main pour ainsi éviter que celles-ci ne semblent rentrer dans la paume et même parfois

³² Cette situation peut se produire lorsque le calibrage n'a pas encore été effectué ou encore si les valeurs maximum repérées par celui-ci ne sont plus valables, lors d'un changement d'utilisateur, par exemple.

en ressortir de l'autre côté.

```
#define M_PI 3.14159265359
#define R60 (M_PI/3)

// Traitement pour le pouce

// Déterminer la quantité de flexion retournée par le capteur
// du pouce
dnewflex=(raw->bFinger[0]-WT5glove_openval[0])/
          (WT5glove_closedval[0]-WT5glove_openval[0]);

// Ajustement pour erreur du capteur
if (dnewflex<0) dnewflex=0.0;
if (dnewflex>1) dnewflex=1.0;
dflex = (oldflex[0] - dnewflex)*R60;
oldflex[0] = dnewflex;

// Rotate the thumb around Z axis
WtObject_rotate(finger[0], Z, dflex, WTFRAME_LOCAL);

// Traitement similaire pour les autres doigts
// for each other finger since thumb==0
for(i = 1; i < 5; i++)
{
    dnewflex=(raw->bFinger[i]-WT5glove_openval[i])/
              (WT5glove_closedval[i]-WT5glove_openval[i]);
    if (dnewflex<0) dnewflex=0.0;
    if (dnewflex>1) dnewflex=1.0;
    dflex = (oldflex[i] - dnewflex)*R60;
    oldflex[i] = dnewflex;
    // Get index to finger segment
    j = (i-1)*3 + 2;
    // Rotate segment around X axis
    WtObject_rotate(finger[j], X, dflex, WTFRAME_LOCAL);
    WtObject_rotate(finger[j+1], X, dflex, WTFRAME_LOCAL);
    WtObject_rotate(finger[j+2], X, dflex, WTFRAME_LOCAL);
}
}
```

Figure 3.1 : Flexion des articulations dans *InterWorlds*

Un autre ajustement concernant la flexion des articulations concerne la rotation de la première phalange du pouce virtuel. Si celle-ci effectue sa rotation dans le même axe que les quatre autres doigts, la main donne beaucoup plus l'aspect d'une patte de primate que celle d'un membre supérieur d'un humain. Pour cette raison, l'angle de flexion du premier segment du pouce doit donc être modifié pour que la rotation s'effectue un peu plus vers l'intérieur de la main. Le code source qui figure dans l'encadré 3.1 illustre bien ce fait ainsi que le traitement permettant de limiter la flexion des quatre autres doigts. L'opération visant le positionnement adéquat du pouce est dans une boucle à part, afin de pouvoir tenir compte de son angle de rotation modifié ainsi que du fait que le pouce comporte une phalange de moins que les quatre autres doigts.

3.1.5 Orientation du gant

Dans la version actuelle du logiciel, étant donné qu'un seul utilisateur à la fois peut se déplacer dans l'environnement, chaque participant n'y est représenté que par la forme de la main virtuelle. Bien entendu, dans une version multiusager, cette représentation ne serait pas adéquate mais, compte tenu de l'utilisation actuelle d'*InterWorlds*, ce choix sembla approprié (voir figure 3.2).



Figure 3.2 : Représentation du gant dans l'environnement synthétique

Dans l'une des versions préliminaires du logiciel, l'image du gant était toujours orientée sur l'axe des Z, vers le positif. Le problème, vite remarqué, apparaissait aussitôt que le participant

tournait la tête de plus de 90 degrés : la main conservait la même orientation par rapport au reste de l'environnement, ce qui avait pour effet direct de grandement diminuer l'effet d'immersion de l'utilisateur. En effet, si l'utilisateur effectuait une rotation de 180 degrés dans l'environnement, il se retrouvait avec une image d'un gant pointant dans sa direction. Il a donc fallu déplacer le point de vue ainsi que le gant lors de chaque rotation et translation de l'utilisateur dans l'environnement afin de conserver l'image du gant en tout temps droit devant le participant.

3.1.6 Détection des intersections

Le premier constat du voyageur virtuel lors de son expérience initiale est l'euphorie qu'engendre le fait de pouvoir se déplacer à travers les objets solides, tels les murs et les plafonds. Dans *InterWorlds*, afin de prononcer l'effet de réalisme lors d'une immersion, il fallait empêcher le participant de traverser les objets de l'environnement. Pour ce faire, il suffisait de tester s'il y avait intersection entre les objets de la simulation et l'utilisateur. Puisque le participant n'est représenté que par un gant, ce test s'avérait plutôt futile. Il fallait donc représenter l'utilisateur par l'image d'un gant jumelée à un prisme invisible englobant les mensurations moyennes approximatives d'un être humain plutôt que par la simple image d'un gant, tel qu'avancé dans la section précédente. Grâce à une telle représentation, il fut possible de détecter les intersections entre le participant (ou en fait son prisme) et le reste du monde.

Afin d'assurer un succès à cette implantation, le participant dans *InterWorlds* est représenté par une pyramide transparente englobant l'image du gant et s'étendant approximativement à la taille du participant, autant en hauteur qu'en largeur. En procédant ainsi, le participant ne peut se retrouver avec sa tête virtuelle à travers le plafond ou même de voir l'image du gant disparaître de l'autre côté d'un mur, sans pouvoir la suivre.

La détection des intersections s'effectue donc en testant, à chaque déplacement du participant, si celui-ci va traverser un élément statique de la scène. Concrètement, il faut, à chaque cycle de la simulation, vérifier si le prisme est intersecté par l'un des objets de la simulation. Lorsque c'est le cas, il suffit de stopper le déplacement et de reculer le prisme (ainsi que le gant et le point de vue de l'utilisateur) de quelques unités. Suite à cette implantation, il s'avéra plus approprié de conserver la position exacte du participant à chaque cycle et de le déplacer à ces coordonnées exactes si, au cycle suivant, il y a intersection. De cette façon, même si l'utilisateur recule à travers un mur, le déplacement palliatif, celui pour l'en faire sortir, n'aggrave pas la situation en l'y enfonçant plus profondément.

L'ajout d'une telle contrainte dans la navigation implique toutefois la résolution d'un nouveau problème : comment le participant peut-il sélectionner un objet si le prisme l'empêche d'y toucher? Deux solutions semblaient alors corriger cette situation. La première technique vise à effectuer une distinction entre les objets statiques (les murs et les plafonds, par exemple) et les objets dynamiques (en l'occurrence, les cadres permettant d'être sélectionnés). La seconde méthode, quant à elle, est beaucoup plus simpliste et ne fait que dupliquer les tests d'intersections : avant de vérifier s'il y a intersection avec un objet de l'environnement, il faut tester si les cadres (ou les autres objets dynamiques), sont intersectés par l'un des sous-objets composant le gant. En effectuant les tests d'intersection dans cet ordre particulier, toute la fonctionnalité du logiciel était conservée et le réalisme de la simulation, préservé.

3.1.7 Conversion du signal vidéo

Un autre problème rencontré lors du développement découle d'une limitation du casque d'immersion de *Virtual i-o*. En effet, ce casque ne supporte qu'une résolution de 640x400 dans un mode de seize couleurs, soit un mode VGA³³ "de base". Une telle capacité est, bien

³³ De l'anglais: *Video Graphics Array*, une norme d'affichage graphique pour les ordinateurs personnels.

évidemment, trop contraignante pour plonger un utilisateur dans un environnement tridimensionnel et espérer qu'il considère l'immersion réussie. Simplement au niveau de la résolution, le 640x400 est nettement insuffisant et si le nombre de couleurs est, par surcroît, limité à seize, il vaut mieux ne même pas songer à atteindre un réalisme convaincant.

Par contre, en plus de ce mode VGA de base, le casque permet l'affichage d'un signal vidéo d'un format alternatif, soit en mode NTSC. Le support pour un tel signal rend le casque compatible avec le signal vidéo d'un téléviseur nord-américain normal, ce qui permet d'arriver à une qualité d'affichage graphique beaucoup plus intéressante que ce qu'offre le mode VGA.

Le mode NTSC constitue donc une excellente alternative puisqu'un tel signal n'est pas tant limité, autant au niveau de la résolution que du nombre de couleurs en entrée, qu'un signal VGA. Le problème se retrouve donc au niveau de la conversion, car une fois le signal VGA de l'ordinateur transformé en NTSC, le casque affichera l'animation interactive sans tenir compte de la résolution et de la couleur originale de l'image VGA. En fait, plus la résolution initiale est élevée, plus la qualité de l'animation sera réaliste. Bien entendu, la résolution maximale est limitée par la capacité de traitement de la station de travail, mais elle l'est aussi par l'équipement de conversion VGA vers NTSC.

Les cartes d'accélération graphiques GLZ-5 qui furent utilisées permettent d'afficher une image comportant plusieurs millions de couleurs dans une résolution supérieure à 1280x1024. Sur le marché actuel, il n'existe pas d'équipement électronique permettant de convertir un signal aussi riche vers une sortie NTSC. En fait, à ce jour, le plus performant des convertisseurs parvient tout juste à transformer un signal SuperVGA 1024x768 vers une telle sortie NTSC.

Suite à une batterie complète de tests subjectifs de qualité, il semblait que cette dernière résolution soit acceptable. Aussi, afin d'assurer un maximum de qualité, la firme privée Téléglobe, avec laquelle le projet s'est déroulé, s'est dotée d'un tel convertisseur de signal. Le coût approximatif de cette boîte de conversion oscille autour de 2500\$, un achat qu'il aurait été impossible d'effectuer sans leur intervention.

3.1.8 Conservation des paramètres

InterWorlds permet actuellement de personnaliser son exécution en fonction de certains paramètres modifiables par l'utilisateur. Ainsi, l'un des paramètres permet de spécifier si un gant numérique est disponible sur le système, un second permet l'affichage d'un écran de présentation personnalisé, etc. Afin de rendre ces informations facilement modifiables, sans avoir à compiler le logiciel pour chaque configuration, il fallait prévoir un mécanisme de conservation des paramètres. De plus, en sauvegardant ces informations sur un support magnétique, ces dernières n'ont pas à être entrées de nouveau à chaque exécution.

Pour ce faire, c'est l'utilisation du *Registry* de *Windows NT* qui semblait la plus appropriée. À chaque exécution, *InterWorlds* vérifie si des paramètres sont disponibles et, le cas échéant, utilise tout simplement des valeurs par défaut et les sauvegarde à l'intérieur du *Registry*. Le tableau 3.3 donne une description de chaque clé du registre spécifique à *InterWorlds*.

Tableau 3.3 : Liste des entrées du Registry		
Nom de l'entrée	Valeur par défaut	Description
Delay	400	Durée de l'immersion, en secondes
BackColor	0x51E	Couleur du fond (5 = quantité de rouge, 1 = quantité de vert E = quantité de bleu)
KeyColor	0x8EE	Couleur pour superposition de la fenêtre vidéo

iGlasses	0	Présence des i-Glasses (valeur booléenne : 0 = faux, 1 = vrai)
Network	0	Fonctionnement en réseau
Joystick	0	Présence d'une manette de jeu
Stereo	0	Utilisation des capacités stéréoscopiques
Glove	0	Présence du Fifth Glove
DefaultServer	SERVER_INET_ADDRESS	Adresse IP du serveur
DefaultFileName	WORLDFILE_NAME	Nom du monde à charger
LastWorld	""	Nom du dernier monde
EndImage	"sunset.tga"	Image marquant la fin de la simulation
StartLogo	"launch.tga"	Image du début
Title	"InterWorlds"	Titre de la fenêtre
DefaultTexture	"sunset.tga"	Texture des murs, par défaut
MPEGViewer	"\\stream\\wnetview.exe"	Chemin de l'application de visionnage des contenus MPEG.

3.1.9 Configuration de l'environnement

La viabilité d'un projet de navigation dans de larges bases de données, tel qu'il est question ici avec *InterWorlds*, dépend grandement de la façon dont l'environnement synthétique peut être créé. En effet, si la création d'une pièce ou des objets en faisant partie exige l'apprentissage d'un nouveau langage de programmation, la popularité du logiciel risque d'en souffrir grandement. Par contre, si la création d'une pièce se fait simplement en précisant la nature et le nombre des objets y siégeant, le même processus de création est grandement simplifié, le rendant ainsi plus attirant pour les concepteurs. Bien entendu, un processus automatique de création d'environnements synthétiques, malgré une simplification évidente de la tâche de conception, introduit indubitablement des limites. En effet, comme tout autre processus automatisé, une telle méthode limiterait la personnalisation des objets de l'environnement (position, taille, etc., des objets par exemple). De plus, un monde créé de la sorte ne pourrait que très difficilement être redistribué ou réutilisé avec d'autres systèmes de navigation.

Lors de la conception d'*InterWorlds*, deux méthodes de définition de l'environnement ont été abordées. La première fut d'utiliser un langage de bas niveau répandu et déjà standardisé alors que la seconde consiste en la création d'un langage de définition permettant de facilement et rapidement définir un environnement. Dans les prochains paragraphes, chacune de ces méthodes sera abordée et les avantages et inconvénients de chacune seront soulignés.

La première méthode, celle visant l'utilisation d'un langage de bas niveau pour décrire un environnement, consiste à permettre aux utilisateurs désireux de créer leur monde de pouvoir le faire en VRML³⁴. Le simple fait que ce langage soit une norme en fait un choix très intéressant : les objets ainsi créés peuvent être récupérés dans d'autres logiciels et *InterWorlds* peut être en mesure de réutiliser les objets créés par d'autres créateurs. Un autre avantage du VRML sur la seconde méthode est la flexibilité qu'il permet. Avec le VRML, il est possible d'adapter chaque élément d'une pièce jusque dans ses moindres détails et même de changer les dimensions ou la couleur de ceux-ci. En annexe F figure un exemple d'un fichier définissant un simple cube sous le format VRML.

Avec un langage d'automatisation, de telles modifications sont impensables, à moins de complexifier le langage afin de lui permettre de telles latitudes. Par contre, le langage d'automatisation, quant à lui, consiste en une description de haut niveau de la scène. C'est un langage très facile à utiliser, contrairement au VRML qui lui, exige une connaissance des fondements de la géométrie tridimensionnelle, de la représentation des points dans l'espace, etc. La simplicité de ce langage résulte en des descriptions d'objets faciles à entretenir et à modifier. Par contre, tel que précisé plus haut, sa faible flexibilité ainsi que le fait que l'on ne puisse redistribuer les définitions de mondes dans des logiciels autres qu'*InterWorlds* en font un choix difficile.

³⁴ Le **Virtual Reality Modeling Language** est une norme permettant de décrire des objets tridimensionnels pour une utilisation à l'intérieur d'un environnement synthétique. La description des objets se fait en spécifiant les points, les arêtes et les surfaces des objets à même un fichier ASCII aisément modifiable avec un simple éditeur de texte.

Pour pallier à de telles lacunes au niveau de la flexibilité, il fut envisagé de paramétrer le langage au point de lui permettre de rendre possible une plus grande personnalisation des objets. Même si cette nouvelle fonctionnalité permet un degré de flexibilité se rapprochant de ce que permet déjà le VRML, le taux de complexité introduit par cet ajout en fait un outil de moins haut niveau. De plus, elle lui fait perdre son avantage de marque, celui d'être simple et facile à utiliser. Cette tentative fut donc rapidement abandonnée.

En dépit des avantages évidents d'un format similaire au VRML, la première version d'*InterWorlds* ne parvient toujours qu'à lire un langage de haut niveau à l'intérieur d'un fichier de description du monde. C'est en fonction des besoins de Téléglobe que le support du VRML fut abandonné. Avec le langage de haut niveau, les dirigeants de l'entreprise pouvaient facilement modifier l'environnement pour représenter le contenu de leur choix. La description d'un tel fichier de configuration, illustrée par un exemple, figure en annexe E.

Suite à l'utilisation continue de cette méthode de configuration, quelques ajouts de fonctionnalité pourraient grandement améliorer les possibilités d'un tel langage. Il pourrait, par exemple, être très intéressant de pouvoir inclure (ou de définir à même le fichier de configuration) des définitions en VRML d'objets génériques comme une table, une lampe, etc. et pouvoir réutiliser ces patrons à l'intérieur du fichier de configuration. Ainsi, suite à l'inclusion d'un fichier de description d'une chaise tridimensionnelle, l'utilisateur pourrait récupérer le modèle de la chaise pour ensuite l'élargir et la changer de couleur, tout cela de façon très simple, à même le fichier de configuration. De tels ajouts auraient comme effet de pouvoir utiliser tout fichier VRML dans l'environnement et ainsi diversifier les expériences de navigation.

3.1.10 Affichage des séquences vidéo

Un aspect primordial de tout logiciel de navigation à travers de larges banques d'informations est la consultation de ces contenus³⁵. Dans le cas d'*InterWorlds*, le contenu est constitué de séquences vidéo résidant originellement sur des bandes magnétiques. Les participants doivent pouvoir, après avoir sélectionné le contenu de leur choix, le consulter de façon aisée et transparente, sans avoir recours à de logiciels externes.

En ce sens, les objectifs d'*InterWorlds* sont particuliers : permettre la consultation interactive d'une séquence vidéo résidant sur un serveur distant. Cet objectif particulier pourrait en fait être décomposé en trois concepts clés, chacun impliquant une précision distincte. Le premier aspect constitue **la consultation de séquences vidéo**, le second, **le serveur distant** et finalement, **la consultation interactive** de ces séquences. Avant d'explicitier le fonctionnement exact des mécanismes retenus dans le cadre de ce projet, il appert opportun d'illuminer chacun des trois sous-concepts précédents.

Le fait que les contenus vidéo soient stockés sur des bandes magnétiques rappelle que ceux-ci sont conservés dans un format analogique. Étant donné le caractère numérique que doit comporter toute information traitée par un ordinateur, il est clair qu'une numérisation devait avoir lieu quelque part dans la séquence des procédures. Cet échantillonnage devait aussi, bien évidemment, se faire en amont du réseau sachant que la bande passante requise pour transmettre un clip vidéo analogique peut facilement dépasser tout genre d'infrastructure de transmission d'information disponible aujourd'hui.

³⁵ En effet, sans la possibilité de pouvoir accéder aux données de la banque, la navigation elle-même devient futile.

Les films composant la base de données ont donc été numérisés en format MPEG et entreposés sur un serveur vidéo dédié. Ce serveur était configuré de façon à permettre plusieurs requêtes simultanées de son contenu sans nuire à sa performance. À partir de ce point, le contenu maintenant numérisé devenait “compatible” avec le monde numérique des ordinateurs et pouvait être beaucoup plus facilement manipulé.

Le second concept à illustrer consiste en la consultation des vidéos provenant d’un serveur distant. Si le contenu était local à la station de consultation, le problème serait aussitôt réglé puisque le contenu est déjà dans un format convenable pour l’affichage sur un moniteur. Mais il en est autrement et il importe alors de définir le mécanisme de transport de cette volumineuse masse d’information. Tel que l’illustre la figure 3.3, les fichiers vidéo peuvent être acheminés à la station cliente dans leur format natif, soit le MPEG, ou alors encodés en un format alors répandu³⁶, le H.320³⁷. Avant d’aborder le troisième concept, celui de la consultation interactive, il est approprié de mieux décrire ces deux scénarios.

La moitié supérieure de la figure 3.3 schématise les diverses étapes impliquées dans la transmission du signal vidéo lorsque celui-ci est numérisé et compressé à l’aide de l’algorithme MPEG. Dans ce scénario, la séquence numérique MPEG est acheminée vers la station cliente à l’aide du réseau Internet. En utilisant le même lien de communication, la station a préalablement indiqué au serveur le nom de la séquence vidéo désirée et le visionnage de cette dernière est assurée par un logiciel dédié, ne requérant aucun matériel

³⁶ Il est important de noter que le projet original a été développé au moment où le MPEG faisait tout juste son apparition sur le marché et la norme H.320 semblait beaucoup plus prometteuse.

³⁷ La norme internationale d’encodage et de compression vidéo H.320 est à la base de tout logiciel de conférence vidéo.

supplémentaire (voir section 2.1.3.1).

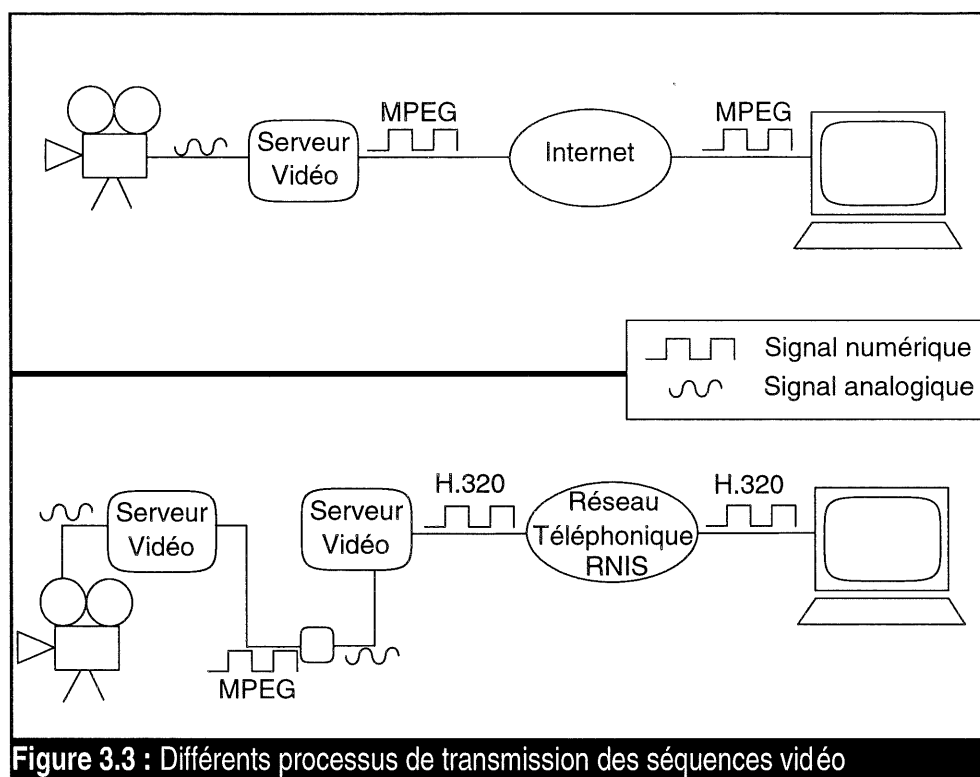


Figure 3.3 : Différents processus de transmission des séquences vidéo

La seconde partie de la figure représente une technique alternative qui fut envisagée au tout début du projet. Dans ce scénario, l'entrée vidéo est encore une fois convertie en format numérique à l'aide du MPEG et les fichiers résultants sont ici aussi entreposés sur le même serveur dédié. Par contre, à partir du moment où le serveur a reçu le nom du fichier requis par la station cliente, la séquence d'action prend une nouvelle tournure. En effet, le fichier MPEG est alors décodé et le signal analogique résultant est ensuite transformé de nouveau selon la norme H.320, puis acheminé à la station cliente via un réseau téléphonique de type RNIS³⁸. Une fois reçu, le signal est décortiqué par des cartes de vidéoconférence puis celui-ci est finalement affiché sur l'écran de la station à l'aide de logiciels conçus à cet effet.

³⁸

Le Réseau Numérique d'Intégration de Services est une technologie de télécommunication permettant de transmettre des données numérique à un débit plus élevé qu'avec un simple modem analogique.

Le degré de complexité de cette avenue est pallié par le haut taux d'infiltration de la technologie de conférence vidéo. En effet, c'est avec une configuration en tout point semblable à celle-ci que **Télélobe Entreprise Média** a mis sur pied un serveur de vidéoconférence sur demande. L'intégration de cette technique à *InterWorlds* comporte cependant quelques problèmes évidents : la décompression du signal de vidéoconférence nécessitait alors des cartes de décompression dédiées ainsi qu'un kit de développement approprié. Il fallait donc acheter le matériel et l'intégrer à la station cliente. Le problème majeur de cette technique est survenu lors des tests préliminaires de faisabilité : ni les cartes de décompression, ni l'environnement de développement n'étaient alors supportés sur la plateforme *Windows NT*. Cette avenue fut donc rapidement abandonnée.

La première méthode, quant à elle, permet de décoder le signal numérique sans avoir recours à des cartes de décompression additionnelles. L'avenue du MPEG est aujourd'hui beaucoup plus prometteuse que celle du H.320 et, si requis, il est aussi possible d'ajouter des cartes de décodage dédiées permettant d'accélérer l'affichage des séquences vidéo MPEG pour ainsi obtenir une qualité d'image plus intéressante. C'est donc cette méthode qui fut retenue, pour tous les avantages qu'elle comporte sur la technique impliquant le H.320.

Étant donné les stricts délais imposés, c'est une application externe de décodage MPEG qui est invoquée pour chaque visionnage. Dans une prochaine version d'*InterWorlds*, il serait très intéressant d'intégrer de telles routines de décompression directement à l'intérieur du logiciel et ainsi permettre l'affichage d'une fenêtre de consultation parfaitement intégrée à même l'environnement tridimensionnel. Le logiciel utilisé est décrit plus en profondeur à la section 2.1.3.1, portant sur les logiciels de consultation vidéo.

Finalement, le troisième concept dont il fut question au début de cette section concerne le degré d'interaction possible lors de la consultation de telles séquences animées. Idéalement, le participant devrait pouvoir manipuler lui-même la totalité des options concernant le

déroulement du film. Cela implique la présence d'une forme de panneau de contrôle virtuel regroupant toutes ces options, accessible à l'intérieur du monde synthétique, à l'aide du gant numérique. Le participant pourrait alors appuyer sur des boutons pour stopper ou redémarrer l'affichage, par exemple. Étant donné l'utilisation d'un logiciel externe pour la consultation, de telles capacités se sont avérées impossibles. Par contre, le démarrage initial des séquences vidéo et l'arrêt sont, quant à eux, déjà disponibles dans la version actuelle. Pour les autres fonctions de consultation comme l'avancement rapide, l'utilisateur doit, dans cette version d'*InterWorlds*, utiliser la souris pour manipuler les boutons de contrôle de l'application de visionnage. Certains ajustements doivent encore être effectués manuellement, mais somme toute, cet aspect d'*InterWorlds* est tout à fait fonctionnel.

3.1.11 Transmission des données

Sachant que les fichiers de description de l'environnement se situent sur un serveur externe et qu'ils doivent être demandés par la station locale de navigation, il fallait maîtriser les mécanismes et les protocoles impliqués lors de la transmission de ces fichiers. Étant donné que le serveur utilisé dans la première phase³⁹ ne requiert qu'un module de transmission des données, c'est un serveur HTTP⁴⁰ type qui fut utilisé. Pour pouvoir tirer profit des capacités d'un tel serveur, il fallait avant tout saisir son fonctionnement ainsi que les multiples fonctionnalités des *sockets*.

InterWorlds englobe donc un module d'interface avec le réseau qui se charge, entre autres, de décortiquer les paquets reçus du serveur et d'obtenir les informations nécessaires, telles que la définition de l'environnement, etc. Toutes les routines de transmission et de réception de

³⁹ La première phase consiste en la navigation dans l'environnement tridimensionnel synthétique par une seule personne à la fois, en empêchant ainsi toute interaction avec d'autres participants et rendant inutiles tous les mécanismes de gestion des interactions dans le serveur.

⁴⁰ Voir section 2.2.4 traitant de la partie serveur de *InterWorlds*.

données provenant du réseau ont été développées en fonction du type de réseau sous-jacent : TCP/IP.

3.1.12 Feed-back tactile

Suite à une immersion prolongée, les participants formulent souvent le même commentaire : “C’est dommage que l’on ne puisse pas ressentir physiquement les objets que l’on touche”. Encore à ce jour, les solutions disponibles pour pallier à cette situation problématique sont, quoique techniquement intéressantes, peu portatives et surtout beaucoup trop dispendieuses (il faut prévoir en moyenne un minimum de 10 000\$ par unité). Ces systèmes consistent en une structure métallique à l’intérieur de laquelle l’utilisateur glisse sa main. Des dispositifs mécaniques ou hydrauliques, selon le cas, sont judicieusement placés et servent à retourner un feed-back tactile à celui-ci.

Les avantages d’un tel périphérique sont multiples : il permet de procurer une sensation de “contact” avec les objets et les murs, il rend possible la manipulation d’objets en permettant de les saisir (de les prendre virtuellement) à l’intérieur du monde synthétique, etc. Mais, en contrepartie de ces avantages évidents, le coût de ces unités en fait des périphériques inaccessibles. Le support pour ce genre de périphérique ne pourra donc être ajouté à *InterWorlds* que s’il est possible d’en obtenir un en prêt, leur prix étant définitivement hors de budget.

3.2 Améliorations futures

Suite à la résolution de ces problèmes, le projet a atteint un certain stade de maturité qui lui a permis de pouvoir être démontré à quelques reprises dans des salons d'envergure internationale. Par contre, suite aux essais en public, certaines lacunes se sont avérées évidentes et quelques ajouts de fonctionnalité seraient souhaitables.

Les prochains paragraphes relèveront quelques-uns des éléments de cette liste d'améliorations souhaitées.

3.2.1 Conversions de pages HTML en site tridimensionnel

Le logiciel *InterWorlds* actuel, même s'il utilise le mécanisme des *sockets* pour communiquer avec un serveur HTTP sur le réseau Internet, ne peut pas interpréter des formats de données autres que le format natif inventé pour ses propres besoins⁴¹. À cet effet, si le produit avait la capacité de construire son environnement à partir d'une description provenant d'un format de fichier plus répandu (par exemple, le VRML), *InterWorlds* pourrait, en plus de pouvoir toujours permettre la consultation de séquences vidéo MPEG à distance, servir de fureteur VRML pour Internet, grandissant ainsi le nombre d'utilisateurs potentiels.

L'implantation de cette fonctionnalité pourrait se faire relativement facilement. Ce n'est que lors de la réception d'un fichier de configuration provenant du serveur HTTP que le logiciel

⁴¹ Voir section 3.1.9 du présent chapitre.

client aurait à interpréter le contenu et aiguiller ce dernier dans l'interpréteur approprié pour poursuivre l'étape de création de l'environnement. Il faudrait, bien entendu, créer un automate permettant de décortiquer les fichiers du nouveau format. Cet interprète pourrait facilement être développé dans un projet complètement distinct pour ensuite être intégré à *InterWorlds* une fois bien testé.

Dans le même ordre de pensée, *InterWorlds* pourrait aussi s'ouvrir à un tout autre marché : celui de la conversion des mondes bidimensionnels en environnements virtuels complets. Connaissant la grandissante popularité des pages HTML sur le WWW⁴², il pourrait être très intéressant de concevoir un interpréteur alternatif capable de convertir des pages HTML bidimensionnelles en pièces ou en concepts tridimensionnels. Cette nouvelle fonctionnalité permettrait fort probablement une nouvelle utilisation d'*InterWorlds*, lui permettant non seulement devenir un outil de navigation complet pour Internet, mais pourrait aussi probablement transformer l'utilisation actuelle que les Internaute font de leur réseau. En effet, s'il est suffisamment complet et rapide, les utilisateurs actuels du réseau préféreront sans doute une navigation tridimensionnelle à une autre qui ne permet que l'affichage de l'information sur deux dimensions, comme le font les logiciels de navigation HTML actuels.

Pour ce faire, suite à une analyse sommaire du format des fichiers HTML, il pourrait s'avérer pertinent de faire correspondre une forme tridimensionnelle particulière à chaque entité du fichier HTML. Ces objets pourraient alors être disposés automatiquement dans un environnement virtuel approprié en utilisant certaines heuristiques permettant de donner une signification particulière à ces objets.

⁴²WWW : *World Wide Web* ou, en français, le réseau global Internet.

Étant donné la tendance actuelle vers la globalisation, le fait de rendre *InterWorlds* ouvert à quelques formats standards d'échange d'informations semble être une évolution naturelle et une transition facilement envisageable.

3.2.2 L'aspect multiusager

Hormis cette ouverture vers des formats de données plus répandus, la tendance actuelle de cette globalisation, qui semble être un élément de grande motivation depuis quelque temps, pourrait être atteinte en rendant *InterWorlds* multiusager. Ainsi, plusieurs participants, provenant des quatre coins du monde, pourraient partager leurs idées sur un sujet donné tout en conservant un contact visuel avec leurs interlocuteurs virtuels. Les utilisateurs pourraient, par exemple, alors qu'ils visitent un musée tridimensionnel et consultent des séquences auditives et visuelles se rapportant aux multiples œuvres exposées virtuellement dans les quelques pièces de l'environnement, échanger leurs impressions et commentaires avec les autres virtuautes aussi présents dans le musée.

Cette complicité, cet échange, n'est actuellement pas disponible sur Internet, mais avec des logiciels multiusager tridimensionnels, une toute nouvelle façon de communiquer s'offre aux navigateurs. Il serait, bien évidemment, très intéressant d'être parmi les premiers à concrétiser un tel produit et ainsi offrir la possibilité de transiger à travers un nouveau média.

La mise sur pied de telles capacités multiusager implique une refonte partielle du logiciel client ainsi que la création complète d'un serveur dédié à l'interaction dans l'environnement. La partie cliente, dans un premier temps, devra, telle qu'elle le fait présentement, permettre de récupérer la description complète d'un monde tridimensionnel virtuel, mais, en plus, devra être en mesure de représenter en temps réel les diverses actions des autres participants présents dans l'environnement. La partie cliente devra donc se charger, à intervalles réguliers,

d'effectuer des requêtes au serveur afin de vérifier si d'autres participants ont agi sur l'environnement (via un déplacement, le déclenchement d'une conversation, la manipulation d'un objet, etc.) ainsi que pour informer le serveur sur la façon dont l'utilisateur local a interagi avec ce même environnement.

Tel que mentionné, tout cet échange d'informations doit se faire en temps réel (soit dans un intervalle de temps inférieur à cent millisecondes) dans un mode de transmission asynchrone sans quoi le délai introduit sera tel que l'utilisateur pourra interagir avec des objets qui n'existent même plus dans l'environnement, au moment où il tente son interaction. En effet, si deux utilisateurs manipulent le même objet et que les actions de ceux-ci sont transmises avec un trop long délai, des disparités entre les versions de l'environnement peuvent surgir pour chacun des participants si aucun mécanisme de vérification de la conformité des mondes n'est disponible. Si le délai de traitement des capteurs est allongé à cause d'une attente prolongée pour de l'information provenant du serveur, le virtuanaute n'en viendra qu'à dissocier ses actions de l'image qui lui est véhiculée et il s'en suivra alternativement une nausée prononcée [14] ou, pis encore, un désintérêt marqué pour la simulation.

Ensuite, après cet échange d'informations, la partie cliente devra effectuer le traitement normal des capteurs de mouvements et finalement procéder à l'affichage sensoriel (audio, vidéo, etc.) des informations récoltées. Ces tâches cycliques se répéteront tout au long de la simulation, à une cadence minimum de dix fois à la seconde, encore une fois afin d'éviter que le participant dissocie ses actions avec le résultat obtenu.

Le serveur, quant à lui, devra être équipé d'un tuyau de communication excessivement rapide et arborant une largeur de bande proportionnelle aux nombre d'utilisateurs anticipés. Les caractéristiques du réseau sont primordiales puisque le serveur devra répondre à toutes les requêtes de tous les clients en temps réel. Son travail consistera à écouter sur un port de communication et fournir, sur demande, les diverses informations requises par les modules

clients. De plus, lorsque des mises à jour du positionnement de certains usagers lui seront acheminées, le serveur devra veiller à mettre à jour une banque de données interne représentant l'état actuel complet de l'environnement. Alors que le module client sera optimisé pour l'affichage de l'environnement, le serveur, quant à lui, sera conçu pour maximiser le transfert des informations vers les multiples clients.

Malgré ces impressionnantes exigences, la mise sur pied de telles capacités d'interaction avec de multiples usagers est très importante étant donné les immenses possibilités qui pourraient en découler.

3.2.3 Amélioration du réalisme

Une autre amélioration fortement souhaitable est celle du réalisme tridimensionnel visuel et auditif. Afin de prononcer l'effet d'immersion dans un environnement virtuel, un des facteurs clés est celui de la qualité du réalisme de l'environnement. Ainsi, des effets lumineux réels, quelques surfaces courbes, la présence de divers matériaux dans la description du monde ainsi que quelques autres effets infographiques impressionnants pourraient rendre une visite dans un monde tridimensionnel virtuel des plus réalistes. De façon similaire, l'ajout de sons et de bruits dont la source est géographiquement identifiable en trois dimensions permettra à coup sûr d'augmenter le réalisme des pièces d'un tel monde.

Pour permettre de tels ajouts, il est par contre primordial de modifier le format initial du fichier de définition du monde, de doter la machine cliente d'accélérateurs géométriques et texturaux ainsi que d'augmenter la vitesse de calcul de cette dernière. Tous ces ajouts requièrent en effet une capacité analytique supplémentaire de la part de chacune des stations clientes et la technologie actuelle ne permet pas facilement de les implanter sans accélérateurs dédiés.

Ces contraintes matérielles supplémentaires impliquent par contre une nouvelle contorsion de la part du logiciel client : la modularité. En effet, le logiciel client doit pouvoir adapter son fonctionnement ainsi que le réalisme de chaque image en fonction de la disponibilité du matériel d'accélération. Par exemple, si un utilisateur possède la toute dernière technologie en matière d'accélération graphique, le logiciel devrait lui permettre une immersion plus avancée en ajoutant le plus d'information visuelle possible, sans tomber sous la limite des quinze images par seconde. De façon similaire, si un autre usager, possédant le même module client, n'a accès qu'à une souris sans aucun autre périphérique et que la capacité de traitement de sa station de travail est limitée, le logiciel devrait pouvoir adapter la simulation en diminuant la qualité des images, par exemple, pour toujours obtenir une animation supérieure au seuil acceptable des quinze images par seconde.

Pour permettre une telle modularité dans la simulation, le fichier de définition du monde doit par contre se complexifier et inclure la notion de degré de réalisme pour chaque élément. Cette information indique au logiciel que, selon la distance séparant l'utilisateur d'un certain objet, l'aspect de cet objet peut être modifié. De cette façon, il est possible d'atteindre une qualité d'image très élevée pour les objets à courte distance du participant tout en n'affichant que l'aspect général des objets plus distants, sans s'occuper des détails. Sans cette information, il devient alors quasi impossible pour le module client de déterminer quelle information est négligeable étant donné une certaine limitation dans les capacités de traitement de la station cliente.

3.2.4 Support pour autres périphériques

Encore avec la même motivation intrinsèque de souhaiter une immersion plus convaincante, il pourrait s'avérer intéressant de permettre à l'utilisateur de s'exprimer à travers une multitude de périphériques de saisie des mouvements. Outre le gant numérique et le casque utilisés jusqu'ici, le support de capteurs plus poussés pourrait grandement augmenter la sensation de

participation dans un environnement tridimensionnel virtuel que pourraient éprouver les participants. En effet, si ces derniers ne sont pas limités à s'exprimer qu'en utilisant un gant et un casque dans le monde virtuel, mais qu'ils puissent aussi par exemple se déplacer en marchant physiquement, en courant ou en pédalant, leur expérience en sera d'autant plus convaincante.

À ce titre, il serait intéressant de créer des points d'entrée et de sortie génériques pour tous les périphériques et ainsi modulariser le logiciel à un plus haut niveau. En effet, si chaque pilote de matériel développé pour le logiciel est conçu autour de points d'interaction prédéterminés (comme par exemple `ETAT_ACTUEL`, `DEPLACEMENT_DIRECTION`, `DEPLACEMENT_QUANTITÉ`, etc.), l'ajout de nouveaux périphériques serait grandement facilité. Une fois le support de nouveaux périphériques ajouté dans *InterWorlds*, il sera plus facile de communiquer avec d'autres participants en utilisant des symboles culturels déjà établis (par exemple, serrer la main d'un ami, faire un sourire et tout le reste du langage gestuel). D'ici là, il serait peut-être intéressant de tenter de trouver d'autres symboles, réalisables avec les périphériques déjà existants, afin de véhiculer ces mêmes messages d'accueil, d'approbation, etc. [11]

Dans ce chapitre, quelques aspects problématiques, certaines difficultés rencontrées ainsi que quelques-unes des améliorations souhaitées pour le logiciel *InterWorlds* furent sommairement abordés. L'une des difficultés énoncées a trait à la stabilisation des capteurs de mouvements ainsi qu'à la technique novatrice qui a permis de filtrer les données reçues par ces périphériques d'entrée/sortie. Le prochain chapitre tentera de clarifier cet aspect du logiciel, tant au niveau de la théorie que sur certains aspects de l'implantation de celle-ci.

CHAPITRE 4

Filtrage par ondelettes

4.1 Introduction

L'objectif de ce chapitre consiste, dans un premier temps, à établir quelques notions fondamentales sur les ondelettes pour ensuite en exposer une application concrète. Loin de se vouloir une référence complète dans le domaine des ondelettes, il sera plutôt question, ici, de démontrer le comportement général de celles-ci pour enfin les appliquer à un problème réel. Donc, après avoir défini certaines bases théoriques et donné quelques exemples d'ondelettes, la situation problématique initiale sera énoncée et la méthodologie utilisée pour pallier au problème sera présentée.

4.2 Aperçu général des ondelettes

L'ondelette est une tendance relativement nouvelle dans la communauté scientifique. En effet, même si les premiers balbutiements des ondelettes virent le jour dans les années 1900, ce n'est que dans les quinze dernières années que l'on s'y attarda avec plus d'intérêt. L'ondelette découle d'une modification de la théorie de Fourier qui tente de représenter des fonctions périodiques satisfaisant des critères stricts en utilisant des combinaisons de fonctions primaires de base : les sinus et les cosinus. Ainsi, en jumelant et en décalant les bases, on parvient à représenter la majorité des fonctions d'un certain type.

Suite à ces conclusions, quelques chercheurs se sont aperçus que la localisation temporelle manquait à la représentation de Fourier. Ceux-ci ont donc créé des fonctions autant localisées en fréquence que dans le temps. Le résultat est éloquent : des fonctions simples et puissantes capables de représenter des fonctions de basses fréquences comportant des fréquences élevées localisées, une astuce impossible avec les transformations de Fourier. Ainsi, des fonctions comportant des discontinuités et d'autres contenant des pics locaux élevés sont mieux approximées à l'aide des ondelettes plutôt qu'avec les transformations de Fourier traditionnelles. Par surcroît, l'approximation requiert substantiellement moins de coefficients en utilisant les ondelettes, permettant ainsi un degré de compression plus élevé.

Utilisées dans des domaines aussi distincts que la physique quantique⁴³, la psychophysiologie et la vision par ordinateur [18], les ondelettes ont profité des études effectuées dans tous ces

⁴³ Grossman et Morlet, dans les années 1980, ont défini les ondelettes en termes de physique quantique.

domaines pour en venir à ce qu'elles sont aujourd'hui : une méthode évoluée de traitement de signal.

4.2.1 Définition des ondelettes

Les ondelettes sont des fonctions ψ définies sur \mathbb{R} qui, lorsque soumises à des translations⁴⁴ et des dilatations⁴⁵, forment un système orthonormal complet pour $\mathbf{L}^2(\mathbb{R})$ ⁴⁶. Elles sont communément notées :

$$\psi_{t,d}(x) := 2^{-d/2} \psi(2^{-d}x - t), t, d \in \mathbb{Z},$$

où t correspond à la translation de la fonction sur \mathbf{R} et d représente un facteur de dilatation. En fait, l'ondelette de base $\psi_{t,d}$ correspond à la version dilatée de la fonction ψ (une fonction ondelette mère), centrée à $t/2^d$. La dilatation de l'ondelette permet la localisation en fréquence alors que les translations permettent la localisation temporelle de la fonction.

Certaines caractéristiques des ondelettes de même que les différences entre certaines d'entre elles seront énoncées dans les prochaines sections.

4.2.2 Types d'ondelettes

Avant de présenter les concepts mathématiques derrière l'analyse par ondelette, il sera question, dans la présente section, des concepts généraux ayant trait à celle-ci. Les

⁴⁴ Ce coefficient permet de déplacer l'ondelette vers la droite ou vers la gauche.

⁴⁵ Il s'agit d'une fonction d'échelle qui permet de varier la période d'oscillation de l'ondelette.

⁴⁶ L'espace de toutes les fonctions de carré intégrable.

caractéristiques globales des ondelettes seront énoncées, puis quelques exemples seront présentés.

4.2.2.1 Propriétés des ondelettes

Voici les huit propriétés souhaitées pour les ondelettes [19]. Bien entendu, il est pratiquement impossible de fusionner toutes ces propriétés dans une même ondelette. C'est pourquoi il en existe plusieurs types, répondant chacune à un besoin particulier, à une application particulière.

- **Orthogonalité** : si les ondelettes sont orthogonales, alors la transformée ondelette rapide (qui sera exposée dans la section 4.2.3.2) est parfaite et la décomposition numérique, stable.
- **Support compact** : la présence de la fonction ondelette dans un intervalle fini est requise lors de l'implantation. Par contre, faute de mieux, il est aussi possible d'utiliser des fonctions à décroissance rapide qui ne seraient pas à support compact.
- **Coefficients rationnels** : dans les implantations, si les coefficients sont des puissances de deux ($d \bmod 2 = 0, d \in \mathbb{N}^*$), la machine pourra les traiter beaucoup plus efficacement.
- **Symétrie** : si la fonction de dilatation et l'ondelette sont symétriques, alors les filtres sont à phase linéaire. Sans cette propriété, on risque d'introduire une distorsion de phase lors de la reconstruction de l'ondelette.
- **Régularité** : l'irrégularité d'une ondelette pourrait nuire à son taux de compression. En effet, afin de compresser un signal, certains coefficients sont fixés à zéro et, si l'ondelette est peu régulière, la qualité de la reconstruction pourrait être faible. De cet argument découle que la régularité des fonctions est plus importante à la décomposition que lors de

leur reconstruction. De plus, davantage de régularité implique une meilleure localisation temporelle des filtres.

- **Nombre de moments nuls** : il détermine le taux de convergence des approximations ondelettes et caractérise aussi la régularité de celles-ci.
- **Expression analytique** : s'il est possible d'obtenir les expressions analytiques de Φ et de ψ , l'analyse harmonique est grandement facilitée. Par contre, de telles expressions ne sont pas toujours disponibles et leur absence ne nuit en aucune façon au travail des ondelettes.
- **Interpolation** : si la fonction d'échelle est discrète, alors il est aisé de la reconstruire à partir d'échantillons puisque c'est précisément ceux-ci qui forment la fonction.

Dans les prochaines sections, il sera question de certains compromis que proposent des classes d'ondelettes.

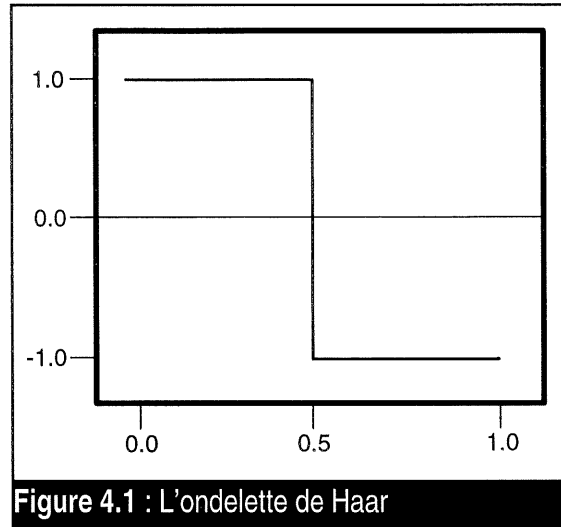
4.2.2.2 Exemples d'ondelettes

La première mention d'une ondelette provient d'une annexe de la thèse d'Haar en 1909. Une des propriétés intéressantes de cette ondelette consiste en son support compact. Cela signifie qu'elle disparaît en dehors d'un intervalle fini⁴⁷. L'ondelette d'Haar ($H := \chi_{[0,1/2)} - \chi_{[1/2,1)}$) est illustrée à la figure 4.1. Cette dernière peut aussi donner naissance à une multitude de fonctions de base à travers les transformations et les dilatations. On obtient ainsi les fonctions

⁴⁷ C'est-à-dire qu'au-delà de la valeur +1.0 et en deçà de -1.0, sur l'axe des réels, la fonction est indéfinie.

$H_{t,d}(x) := 2^{-d/2} H(2^{-d}x - t)$, $t, d \in \mathbb{Z}$ où t et d correspondent aux coefficients de ces transformations et dilatations, supportées sur l'intervalle $\left[\frac{t}{2^d}, \frac{t+1}{2^d}\right]$.

L'ondelette de Haar n'a qu'un seul moment nul. Si on effectue une projection d'une fonction sur cette base, la projetée aura l'allure d'une fonction en escalier. Pour plus de régularité, il faudrait utiliser des fonctions comportant plus de moments nuls.



Parmi les autres types d'ondelettes, on peut retrouver la famille des Daubechies, créée par Ingrid Daubechies, depuis une dizaine d'années. Une ondelette de Daubechies ayant M moments nuls vérifie l'équation de dilatation suivante :

$$\begin{aligned}\phi(x) &= \sqrt{2} \sum_{t=0}^{2M-1} h_{t+1} \phi(2x - t) \\ \psi(x) &= \sqrt{2} \sum_{t=0}^{2M-1} g_{t+1} \phi(2x - t)\end{aligned}$$

où les coefficients h_t et $g_t = (-1)^t h_t$ et cela pour tout $t = 1, 2, \dots, 2M$.

Les différents membres de cette famille se distinguent entre eux précisément par le nombre de moments nuls [20] : D1 correspond à une Daubechies à un seul moment nul (identique à l'ondelette de Haar), D4 est une Daubechies à 4 moments nuls, etc. Les ondelettes de cette famille sont consistantes, orthogonales et régulières. Elles satisfont aussi une propriété d'approximation.

Parmi les autres familles, on retrouve les Coiflets, qui sont en fait un cas particulier des Daubechies, les ondelettes biorthogonales et les pseudo-Coiflets [21].

Maintenant que les caractéristiques importantes des ondelettes ont été présentées, il semble approprié de déterminer les différences entre quelques-uns des principaux types d'ondelettes.

Tableau 4.1 : Comparaison de quelques familles d'ondelettes							
Familles d'ondelettes	Support compact		Expression analytique		Symétrie	Orthogonalité	
	Primaire	Dual	Primaire	Dual		Semie	Totale
Daubechies	Oui	Oui	Non	Non	Non	Oui	Oui
Biorthogonale spline	Oui	Oui	Oui	Non	Oui	Non	Non
spline semi-orthogonale	Oui	Non	Oui	Oui	Oui	Oui	Non
Meyer	Non	Non	Non	Non	Oui	Oui	Oui
spline orthogonale	Non	Non	Oui	Oui	Oui	Oui	Oui

4.2.2.3 Différences entre les types d'ondelettes

Le tableau 4.1 tiré de [19] illustre les compromis proposés par cinq familles d'ondelettes. Même si certaines familles n'ont pas été présentées à l'intérieur de ce chapitre, le tableau devrait permettre au lecteur de choisir parmi les familles selon les besoins d'une application précise. Pour plus de détails sur les familles, voir [19].

4.2.3 Analyse par ondelettes

Alors qu'il serait aussi possible d'aborder les ondelettes à travers la transformation ondelette continue, il ne sera ici question que de l'analyse multirésolution, telle que conçue par Ingrid Daubechies [22].

Conceptuellement, l'analyse multirésolution consiste à effectuer les transformations en plusieurs étapes. À chacune de ces étapes, il faudra concentrer l'analyse sur une partie plus petite du signal, un sous-espace, de la même façon qu'un zoom agrandirait une image photographique. Ainsi, l'analyse de la partie agrandie du signal permettra d'en conserver les détails et éviter, de la même façon, que l'approximation vienne à perdre certains détails discernables seulement lors d'un agrandissement. L'énergie totale du signal est conservée, sachant que l'union des fonctions appliquées sur les sous-espaces converge vers la fonction originale.

Mathématiquement, les fonctions « **échelle**⁴⁸ » et « **ondelette** » sont définies comme suit :

$$\begin{aligned}\phi(x) &= 2 \sum_t h_t \phi(2x - t) \\ \psi(x) &= 2 \sum_t g_t \phi(2x - t)\end{aligned}$$

Tel qu'illustré précédemment, c'est au moyen des translations et des dilatations de l'ondelette mère que l'on parvient à créer une base orthogonale : l'ondelette de base. Si l'on utilise la définition mathématique d'une ondelette énoncée plus haut, et que l'on se limite à des coefficients m et n choisis dans une grille discrète, on obtient que $\psi_{m,n}(x) = a_0^{-m/2} \psi(a_0^{-m} x - nb_0)$.

⁴⁸

Aussi appelées équation de dilatation, ou équation de raffinement

Si, par surcroît, dans le but de simplifier l'équation, on fixe $a_0=2$ et $b_0=1$, on parvient à une forme simple correspondant à une base orthonormale :

$$\psi_{m,n}(x) = \frac{\psi(2^{-m}x - n)}{\sqrt{2^m}}$$

Ensuite, afin d'étendre le domaine de l'ondelette à différentes résolutions, on doit utiliser une équation de dilatation $W(x)$:

$$W(x) = \sum_{t=-1}^{N-2} (-1)^t c_{t+1} \psi(2x + t)$$

où c_t sont les coefficients de l'ondelette. C'est précisément cette fonction qui permet d'appliquer l'ondelette de base à différentes résolutions du signal afin de bien récupérer les hautes fréquences localisées qu'il contient. Les coefficients de l'ondelette doivent, pour ce faire, satisfaire les contraintes linéaire et quadratique suivantes :

$$\sum_{t=0}^{N-1} c_t = 2, \quad \sum_{t=0}^{N-1} c_t c_d = 2\delta_{d,0}$$

où δ est la fonction de variation et d , l'index de position. Les équations précédentes permettent de mieux comprendre la nature mathématique des ondelettes. On remarque aussi qu'avec un tel système, il devient facile d'utiliser une ondelette différente, ou mieux adaptée au signal en entrée, pour obtenir de meilleurs résultats, ou tout simplement afin de comparer les performances de chacune. En effet, la modularité du système ci-dessus est telle, que si on remplace ψ par l'ondelette de Haar ou une autre de la famille de Daubechies⁴⁹, les transformations sont encore applicables et les équations, encore valides.

⁴⁹

Se référer à la section 4.2.2.

Les deux prochaines sections traiteront d'expansions par ondelettes dans un domaine discret. La transformée ondelette discrète sera initialement définie puis, dans un second temps, il sera question d'un algorithme de décomposition/recomposition avancé par S. Mallat : la transformée ondelette rapide.

4.2.3.1 Transformée ondelette discrète (DWT⁵⁰)

Alors qu'en théorie les transformations ondelettes continues *peuvent* être plus intéressantes, dans la pratique, puisqu'un échantillonnage est requis afin de pouvoir traiter des données, le signal en entrée est discret. Voilà pourquoi il est préférable de faire appel à une analyse discrète dans une telle situation.

Une transformée ondelette discrète, notée T , associe une fonction f à l'ondelette de base $\psi_{d,t}$ vue précédemment :

$$(Tf)_{m,n} = \langle \psi_{m,n}, f \rangle$$

où $\langle x, y \rangle$ correspond au produit scalaire de x par y et où m , le paramètre d'échelle et n , le paramètre de translation, $\{m, n\} \in \mathbb{Z}^2$. De plus, si certaines conditions d'admissibilité sont satisfaites, alors il est possible de reconstruire f à partir de ses coefficients d'ondelette. Par contre, la reconstruction ondelette sera couverte plus en détail dans la prochaine section.

Ainsi, il est donc possible de transformer un vecteur ou un signal quelconque à l'aide des ondelettes pour obtenir un vecteur intermédiaire w . Puis, après avoir utilisé ou manipulé ce vecteur, de le reconvertir en ce qu'il était avant la transformation initiale. Cette particularité

⁵⁰

En anglais : *Discrete Wavelet Transform*.

de pouvoir revenir à la fonction originale rend des exercices tels la compression ou le filtrage de signaux très efficaces.

Afin de mieux comprendre le comportement de la transformation ondelette discrète, il peut être intéressant de considérer les coefficients c_i comme un filtre. Celui-ci est placé dans une matrice de transformation qui est appliquée à un vecteur, le signal en entrée. Les coefficients sont ordonnés selon deux modèles : le premier agit en tant que filtre atténuateur et le second tente de faire ressortir les détails des données. Conjointement, les coefficients font ressortir, après un certain nombre d'itérations sur le vecteur initial, l'énergie du signal. Et c'est à partir du vecteur de sortie w que l'on peut, avec une transformation inverse, obtenir le signal original, intact.

La matrice des coefficients est appliquée à l'intrant avec un algorithme hiérarchique appelé **algorithme pyramidal**. Dans la matrice, les coefficients sont arrangés de façon à ce que les rangées impaires contiennent les coefficients de l'ondelette qui agiront comme filtres alors que les rangées paires, quant à elles, hébergeront un arrangement des coefficients qui permettra de faire ressortir les détails du signal. Une fois la matrice appliquée au vecteur original, celui-ci est lissé et décimé de moitié puis, on lui applique la matrice de nouveau. On répète le processus à plusieurs reprises, jusqu'à ce que les coefficients du vecteur obtenu ne soient plus significatifs⁵¹. Ainsi, l'application de la matrice fait à la fois ressortir une résolution plus élevée des données et lisse les données de l'itération (résolution) précédente. Le produit final consiste en un vecteur lissé ainsi que de la totalité des vecteurs de détail accumulés [23].

⁵¹ À ce point-ci, si le processus est répété, les coefficients introduisent du bruit.

4.2.3.2 Transformée ondelette rapide (FWT⁵²)

La transformée ondelette rapide œuvre, elle aussi, dans un domaine discret. Elle vise à décomposer la transformée ondelette discrète en un produit de matrices creuses. La raison est simple : les vecteurs produits par la DWT ne sont pas creux et cela entraîne de la complexité dans les traitements. En travaillant avec des matrices creuses, le coût du traitement est diminué à $O(n)$.

Il est approprié d'utiliser l'analyse multirésolution pour décomposer une séquence discrète $c^0 = (c_n^0), n \in l^2(Z)$. Ainsi, après quelques décompositions qui sont illustrées dans [19], on peut trouver que

$$c_k^1 = \sum_n h(n-2k)c_n^0 \quad \text{et} \quad d_k^1 = \sum_n g(n-2k)c_n^0,$$

avec

$$h(n) = 2^{-1/2} \int \phi(x/2) \phi(x-n) dx$$

$$g(n) = 2^{-1/2} \int \psi(x/2) \phi(x-n) dx$$

En définissant H et G comme opérateurs, on obtient :

$$(Ha)_k = \sum_n h(n-2k)a_n$$

$$(Ga)_k = \sum_n g(n-2k)a_n$$

⁵²

En anglais : *Fast Wavelet Transform*

Puis, lors de la reconstruction, il suffit d'appliquer l'opération inverse. En effet, afin d'obtenir une reconstruction parfaite, il suffit que $\tilde{H}H + \tilde{G}G = I$. Pour trouver les coefficients, on peut alors utiliser l'équivalence suivante :

$$\begin{aligned} c_n^{j-1} &= \sum_k h(n-2k)c_k^j + \sum_k g(n-2k)d_k^j \\ &= \tilde{H}c^j + \tilde{G}d^j \end{aligned}$$

Maintenant que les concepts mathématiques ont été énoncés, il apparaît plus concret de représenter cette transformation à l'aide d'un algorithme. L'encadré de la figure 4.2 illustre comment il est possible de décomposer et de recomposer un vecteur de taille n à l'aide de la FWT.

L'algorithme est particulièrement simple. Il s'agit en fait d'appliquer H , puis d'appliquer le filtre G au signal original. Ensuite, il suffit de sous-échantillonner le vecteur en entrée (par exemple, de ne conserver que les valeurs ayant un indice pair). Pour la reconstruction, il s'agit de sur-échantillonner (ajouter un zéro entre chaque échantillon) et ensuite effectuer le filtrage [19]. Suite à l'exécution de la procédure de décomposition, le vecteur w contient les coefficients ondelette finaux :

$$w = \left[s_0^0 \ d_0^0 \ d_0^1 \ d_1^1 \ d_0^2 \ \dots \ d_3^2 \ \dots \ d_0^{n-1} \ \dots \ d_{2^{n-1}-1}^{n-1} \right],$$

correspondant aux coefficients de lissage suivis d'une série de coefficients exprimant les détails du signal traité.

4.3 Description initiale du problème

Après avoir abordé certains aspects théoriques des ondelettes, il apparaît approprié, à ce point, de présenter une application concrète de l'utilisation de ces fonctions. Dans l'exemple suivant, les ondelettes furent incorporées à un projet déjà existant dans le but d'améliorer la qualité d'un signal unidimensionnel en entrée.

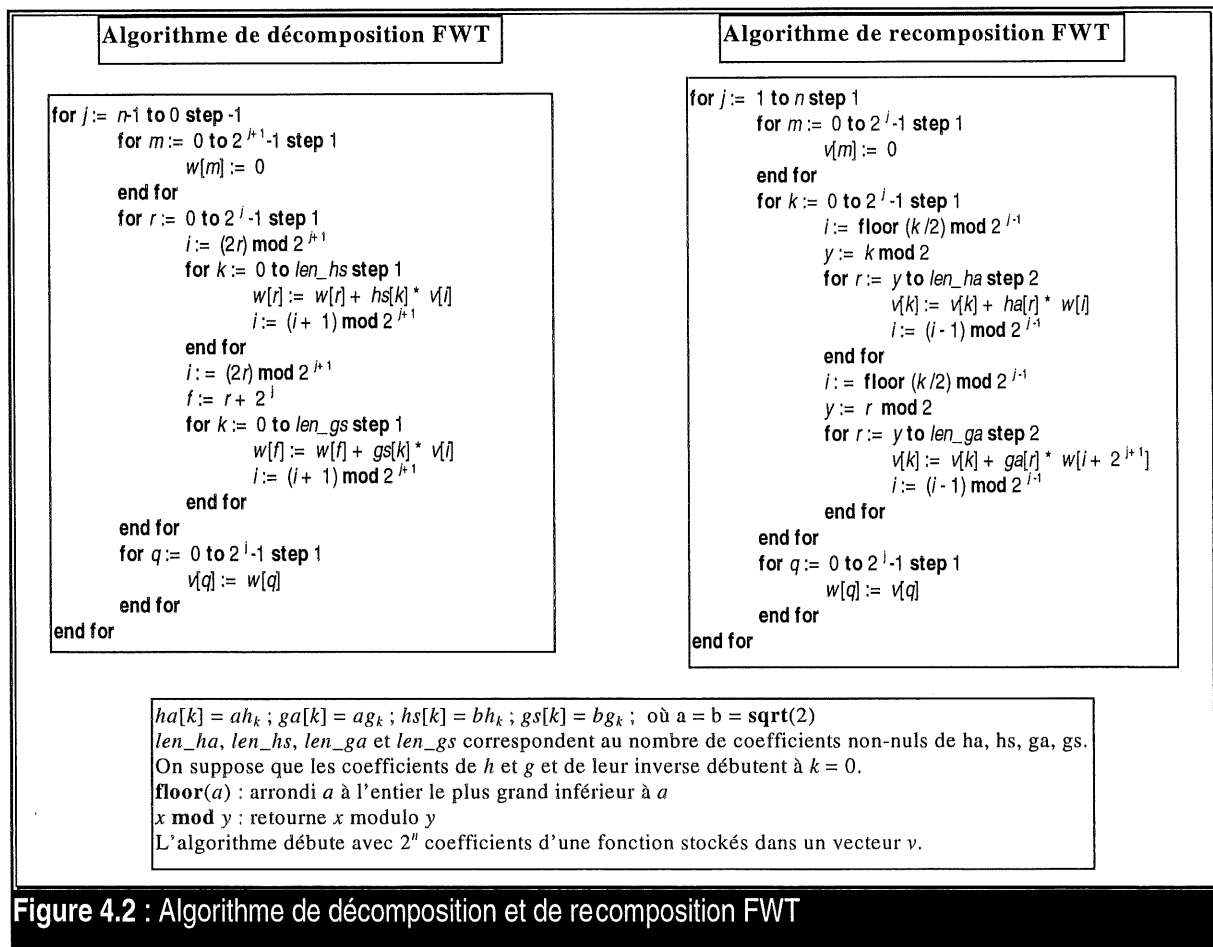


Figure 4.2 : Algorithme de décomposition et de recomposition FWT

4.3.1 Description du projet

Tel que décrit dans le chapitre précédent, l'utilisation des ondelettes à l'intérieur de ce projet a pour but ultime de purifier le signal provenant des capteurs de mouvements du casque d'immersion. L'arrangement initial incorporait des périphériques de bas de gamme qui ne produisaient que des signaux impurs, comportant beaucoup de bruit, tel que le démontre la figure 4.3. Il en résultait ainsi des mouvements non souhaités par le navigateur et, suite à une utilisation prolongée, l'apparition du syndrome du navigateur virtuel (maux de cœur, etc.) se faisait sentir.

Le capteur de mouvements relié au casque fournit, sur demande, des informations sur le positionnement de la tête de l'utilisateur à raison de 250 Hz (la direction de la tête du participant est analysée 250 fois par seconde). Les triplets envoyés correspondent à l'orientation de la tête autour de l'axe des X, celui des Y et aussi des Z⁵³. Les composants X et Z du triplet obtenu sont des entiers compris entre -127 et 127 correspondant à la valeur absolue du capteur autour de chacun des axes. Le troisième élément du triplet, tel que décrit à la section 3.1.1, représente la quantité de rotation autour de l'axe des Y. Puisque le capteur permet d'effectuer des rotations complètes autour de cet axe, les valeurs retournées sont incluses dans l'intervalle]-180..180[.

L'aspect d'instantanéité associé à l'utilisation d'un capteur exige de l'application, afin qu'elle puisse utiliser de façon efficiente ce dernier, qu'elle traite les informations obtenues en temps réel (ou dans un temps relativement court) sans quoi le délai qui s'introduit entre un

⁵³ En anglais : respectivement, le *pitch*, le *yaw* et le *roll*.

déplacement de la tête et le déplacement de l'image peut potentiellement provoquer des nausées chez le navigateur.

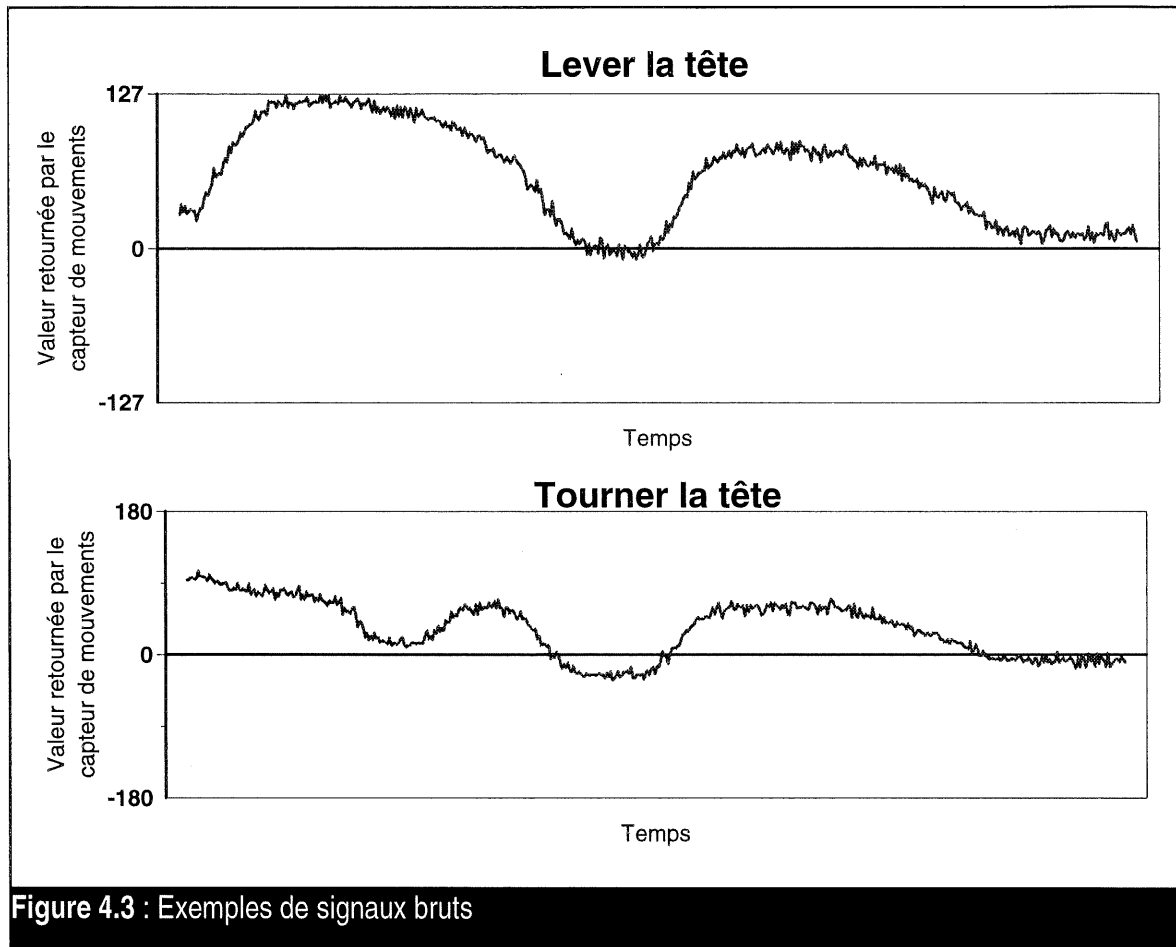


Figure 4.3 : Exemples de signaux bruts

4.3.2 Situation souhaitée

Bien entendu, il s'agissait ici de stabiliser le signal brut provenant du casque, puisque la qualité de celui-ci était affectée par la présence de bruit. Les ondelettes devaient donc, en quelque sorte, filtrer le signal en entrée afin d'éliminer le bruit présent.

Tout le traitement impliqué par ce filtrage ne doit, en aucune façon, ralentir la vitesse à laquelle les signaux du capteur sont interprétés. Le tout doit donc se faire en temps réel, ou, du moins, n'introduire aucun délai susceptible de ralentir le traitement au-delà de la barrière des 100 ms⁵⁴.

4.4 Méthodologie et processus d'implantation

Dans les prochaines sections, il sera question de la façon dont les ondelettes ont pu améliorer la situation du capteur de mouvements trop sensible. L'idée générale de la transformation sera énoncée puis, l'intégration du code sur les ondelettes à celui de l'application de réalité virtuelle sera décrite.

4.4.1 Description des processus impliqués

Afin de filtrer le signal en entrée à l'aide des ondelettes, il fallait soit développer une librairie de transformées ondelettes ou, plus simplement, en utiliser une déjà existante. C'est la seconde option qui fut retenue étant donné l'omniprésence des contraintes temporelles. Le choix fut arrêté sur le package **WaveLib** de l'IRISA⁵⁵. Une fois la compilation de la librairie complétée, quelques modifications ont dû être apportées quant à la façon dont le signal était traité. En effet, puisque le capteur de mouvement relié au casque retourne continuellement des informations (en temps quasi-réel, à une cadence de 250 Hz), et que la librairie est construite

⁵⁴ 100 ms correspond environ au délai maximal toléré par un navigateur virtuel entre le déplacement d'un capteur (gant, casque, etc.) et la réaction de l'environnement face à ce stimulus.

⁵⁵ L'Institut de Recherche en Informatique et Systèmes Aléatoires, à Cedex en France.

de façon à traiter un signal fixe déjà disponible, un mécanisme de tamponnage a dû être introduit.

Le fonctionnement de ce mécanisme, aussi illustré dans la figure 2.2 est décrit dans les prochains paragraphes.

Le casque retourne la position absolue de la tête de l'utilisateur sous la forme de trois entiers : l'orientation autour de l'axe des X, celui des Y et celui des Z. Chacune de ces valeurs est insérée à la fin d'une liste contenant les seize dernières lectures provenant du casque. À chaque itération, le vecteur perd sa plus *vieille* valeur au dépend d'une toute récente, originant de la lecture, tout ça un peu à la manière d'une liste circulaire. Le vecteur est ensuite *aplati* et décompressé par une transformée ondelette rapide. Le vecteur qui en sort correspond aux coefficients de l'ondelette représentant le signal ainsi que le bruit à enlever.

Ensuite, un seuillage est appliqué au vecteur. Cette technique est explicitée un peu plus loin dans cette section.

La recomposition ondelette est ensuite appliquée sur le vecteur à la sortie du seuillage et le vecteur de sortie contient finalement les positions absolues du capteur auquel le bruit inhérent a été retranché. Il suffit ensuite de saisir la tête du vecteur et de la fournir à l'application hôte afin qu'elle puisse transformer les objets de l'environnement en fonction des données du capteur.

Le fait d'utiliser quelques valeurs antécédentes aux orientations courantes obtenues du capteur permet de traiter la position courante en fonction des derniers mouvements de l'utilisateur. Il serait ici peut-être intéressant d'utiliser les valeurs filtrées plutôt que les valeurs brutes dans le

vecteur circulaire. Les valeurs ainsi obtenues pourraient alors se filtrer d'elles-mêmes une seconde fois et ainsi s'épurer de façon plus efficace du bruit inhérent.

Une seconde complication mineure s'est introduite dans le développement : la librairie permet de compresser un signal, mais non de le filtrer. Une fonction de seuillage a ainsi dû être incorporée de la même façon, afin de permettre cette nouvelle fonctionnalité.

Quelques techniques de seuillage sont décrites dans [24]. On y décrit le seuillage unilatéral qui consiste à adopter une politique de « *keep or kill* ». Il suffit, dans ce cas, de comparer la valeur absolue d'un coefficient d'ondelette à une valeur arbitraire λ . Si l'amplitude du coefficient est inférieure à λ , alors le coefficient est remplacé par 0. On peut représenter le seuillage unilatéral par :

$$d_{jk}^{hard} = \begin{cases} 0, & d_{jk} < \lambda \\ d_{jk}, & d_{jk} \geq \lambda \end{cases}$$

Le seuillage unilatéral est avantageux lorsqu'on est intéressé à obtenir le plus petit code d'ondelette. En effet, il provoque de longues séquences de zéro et celles-ci peuvent être codées de façon efficace.

C'est cette même technique qui fut utilisée pour compenser les lacunes du casque d'immersion au niveau de la stabilité des capteurs. En effet, le signal provenant du capteur de mouvement du casque introduisait du bruit et le résultat, tel que mentionné auparavant, se traduisait par un mouvement saccadé continu, même lorsque le casque était stabilisé sur une table. Ainsi, avec la méthode du seuillage unilatéral, le remplacement de certains coefficients par des zéro est interprété comme un mouvement nul, équilibrant ainsi le capteur.

Les autres méthodes de seuillage introduites dans l'article sont le seuillage graduel, le seuillage par quantité et le seuillage universel. Ces méthodes ne seront pas décrites dans ce chapitre puisque, dans le cadre du projet actuel, celles-ci se sont avérées moins efficaces que la première.

Hormis le seuillage et la transformation des informations du capteur en vecteur, aucune autre adaptation ne fut requise dans la librairie *WaveLib*.

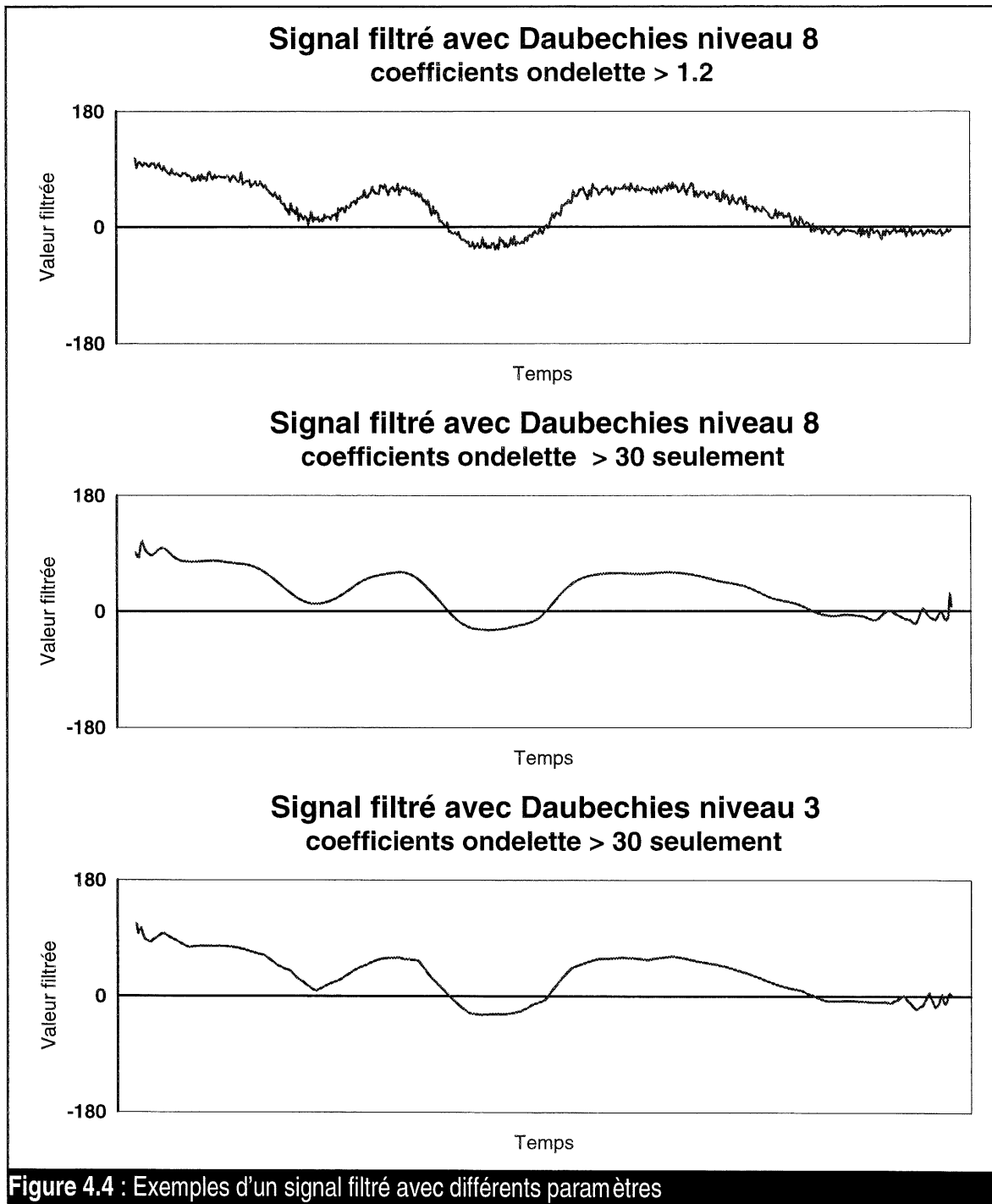
Dans la prochaine section, il sera question de l'intégration du code de la librairie à celui de l'application hôte. Puis, dans une section ultérieure, une description des résultats sera présentée.

4.4.2 Intégration du code dans l'application *InterWorlds*

En annexe B figurent les quelques fonctions qui furent requises pour utiliser les fonctionnalités incluses à l'intérieur de la librairie *WaveLib* pour la création de l'application originale.

Le programme de navigation dans un environnement tridimensionnel permet l'utilisation d'une fonction externe pour obtenir des informations des divers capteurs reliés au système. Dans le cas présent, c'est la fonction `HeadTrackerUpdateFonction` qui effectue cette tâche. Dans le corps de la fonction, on interroge le capteur pour obtenir les trois orientations puis, on les transmet à la fonction de filtrage : `Filter`. Cette dernière crée un vecteur à la façon expliquée plus haut, crée les filtres et effectue la décomposition. À chaque itération et pour chacune des trois valeurs obtenues du capteur de mouvement, le seuillage survient entre la décomposition et la recomposition des coefficients.

Le cycle complet se termine lorsque la fonction `HeadTrackerUpdateFonction` extorque la dernière valeur du vecteur et la rend disponible au reste de l'application via un appel à la fonction `WTsensor_setrecord`.



4.4.3 Description des résultats

L'implantation d'une telle technique de filtrage implique inévitablement l'adoption de certains concepts, de certaines valeurs, afin d'assurer un fonctionnement optimal de l'algorithme. Les prochains paragraphes serviront à décrire la logique ainsi que les motivations qui ont mené à l'acceptation de ces choix.

Avant même d'effectuer le filtrage du signal, chaque valeur obtenue du capteur est déposée dans un tampon. Afin de ne pas surcharger l'ordinateur avec un processus de filtrage coûteux en terme de traitement, il fallait tenter de maintenir la taille de ce tampon le plus petit possible. La valeur choisie fut obtenue en tentant de maximiser la grandeur du tampon afin d'obtenir un meilleur filtrage tout en veillant à la minimiser pour ne pas engendrer une charge supplémentaire sur le processeur. L'expérimentation justifia donc un tampon de taille 16: cette valeur s'avéra adéquate autant au niveau de la qualité du signal filtré qu'au plan de la quantité de traitement requise pour l'obtenir.

De plus, afin d'assurer un filtrage efficace du signal en entrée, le logiciel s'assure que chaque valeur obtenue du capteur est positionnée à l'intérieur d'un tampon circulaire. L'utilisation d'une telle technique permet de limiter l'effet négatif du filtrage sur les extrémités d'un échantillon. En effet, l'une des lacunes du filtrage d'un signal à l'aide d'ondelettes réside dans la représentation des bornes du signal en sortie. Cet effet peut facilement être remarqué dans la figure 4.4, où les extrémités de chacun des signaux filtrés ne correspondent en rien au signal original.

Tel que décrit dans la section 4.4.1, le tampon circulaire n'est ni plus ni moins qu'un vecteur de taille 16 avec un pointeur indiquant l'endroit où l'on doit insérer les données. Le pointeur est déplacé à chaque itération et lorsqu'il atteint une extrémité du tampon, il revient simplement à l'origine de celui-ci.

Une fois le vecteur empli de données du capteur, il est aussitôt compressé à l'aide des ondelettes. Le résultat de cette opération correspond à une série de valeurs représentant les coefficients de l'ondelette. Ceux-ci sont alors filtrés avec une fonction de seuillage qui a pour objectif de libérer la série des coefficients trop faibles. Suite à une telle compression, l'énergie globale du signal peut être adéquatement représentée par quelques coefficients seulement. Les autres valeurs, plus faibles celles-là, correspondent au bruit inhérent au signal en entrée. L'annulation systématique de ces valeurs, en les fixant à zéro, permet de conserver le signal original, tout en filtrant les éléments indésirables.

Dans *InterWorlds*, le seuil minimum pour les coefficients de l'ondelette fut fixé à 30, une valeur qui s'est révélée adéquate pour une utilisation dans un logiciel requérant une interaction en temps réel. Lorsqu'un seuil plus conservateur était utilisé, par exemple 5, le signal conservait, malgré le filtrage, trop de bruit, résultant en une vibration tout autant présente que dans le signal brut. Avec une constante plus agressive, le signal filtré s'aplanissait et le déplacement dans l'environnement devenait très ardu.

Les deux premiers graphiques de la figure 4.4 démontrent de façon assez évidente la différence entre la fixation d'un seuil trop permissif (1,2 dans le cas du premier graphique) versus un autre plus approprié (30, pour le second graphique). Malgré l'évidente amélioration du signal en utilisant un seuil plus élevé, la séquence obtenue suite à un filtrage avec un seuil de 30, tel que le démontre le graphique du milieu de la figure 4.4, laisse encore à désirer au niveau du réalisme, lorsque vient le temps de déplacer l'environnement virtuel en fonction d'un tel signal filtré. Effectivement, le signal ainsi produit est démuné de tout bruit, mais sachant que les changements de direction de la tête de l'utilisateur se font habituellement par mouvements brusques et saccadés, un signal filtré de la sorte ne permettrait pas de bien représenter une telle situation. Voilà pourquoi certaines modifications ont dû être apportées au type d'ondelette utilisé.

Le troisième graphique de la figure 4.4 correspond au signal filtré à l'aide d'une ondelette de type Daubechies 3. Une telle ondelette, qui ne comporte que trois moments nuls, produit un signal beaucoup moins "harmonieux" que celui filtré à l'aide d'une Daubechies 8, par exemple. En effet, le signal comporte des pics et des pointes qui ne se retrouvent pas avec une ondelette comportant un nombre supérieur de moments nuls. Par contre, dans une application du type dont il est question ici, c'est précisément ce type de comportement qui est recherché. Lorsqu'un utilisateur navigue dans un environnement tridimensionnel à l'aide d'un casque d'immersion, les mouvements de sa tête sont saccadés plutôt que graduels et le signal filtré avec une ondelette comportant peu de moments nuls conserve cet aspect de spontanéité des déplacements. De plus, les commentaires recueillis lors de l'expérimentation confirment qu'un tel filtre correspond mieux à la fonctionnalité recherchée.

Tel que décrit dans les chapitres précédents, la puissance de la machine utilisée dans le cadre de ce projet permettait très facilement d'effectuer le traitement supplémentaire imposé par le filtrage du signal à l'aide des ondelettes. Somme toute, les quelques cycles d'exécution requis lors du filtrage ne nuisaient guère à la nature interactive d'une telle simulation en temps réel et permettaient de parvenir à une immersion plus réaliste en éliminant les effets de certaines contraintes matérielles.

4.5 Conclusion

Dans ce chapitre, il a été question de l'intégration du code de la librairie *WaveLib* à une application déjà existante. La transition s'est faite sans trop d'embûches, mais c'est surtout au niveau de l'adaptation du signal en temps réel que l'adaptation a dû s'effectuer. Une technique de tamponnage sur pseudo liste chaînée a été introduite et c'est cette dernière qui fut

testée. Le résultat est encourageant, mais encore beaucoup d'ajustements seraient requis afin d'en faire un outil puissant.

La prochaine étape consisterait à essayer d'autres types d'ondelettes lors de la transformation et de comparer celles-ci afin de déterminer laquelle serait la plus appropriée à ce genre de filtrage. Il serait intéressant de pouvoir modifier dynamiquement l'ondelette utilisée lors des transformations afin de pouvoir constater de façon interactive les comportements modifiés. Somme toute, l'implantation est concluante et sera probablement utilisée dans une prochaine version du logiciel.

CONCLUSION

Le logiciel *InterWorlds* a été initialement développé dans le but d'utiliser les avantages de l'immersion dans un environnement virtuel pour naviguer dans une large banque de données. Pour satisfaire certaines contraintes imposées par le milieu privé, la consultation des objets catalogués dans cette base de données devait générer un trafic substantiel sur des liens de communication. Précisément pour parvenir à cette fin, c'est une base de données contenant de multiples séquences vidéo qui fut construite. L'application finalisée permet donc de rechercher efficacement à l'intérieur d'un environnement tridimensionnel des séquences vidéo parmi un vaste ensemble de films numérisés.

La *virtualisation* des données, ou la disposition de celles-ci dans un monde synthétique virtuel, requiert un certain effort d'organisation pour permettre une navigation intuitive à travers les objets de l'environnement. Puisque le développement d'*InterWorlds* a été axé sur l'aspect de la navigation plutôt que sur l'organisation des éléments de la banque d'information dans le monde, cette étape de pré-traitement n'a pu être complétée. Hormis cet aspect, qui aurait certainement bénéficié d'une forme d'automatisation, le développement d'*InterWorlds* est bien avancé : la première version du logiciel est fonctionnelle, incluant la navigation et la consultation des contenus vidéo.

Suite à son achèvement, le logiciel *InterWorlds* a été formellement testé à deux reprises, dans le cadre d'expositions internationales. La première, qui a eu lieu en Autriche, a permis de tester et de finaliser la mise au point de l'aspect de consultation des séquences vidéo à distance en véhiculant ces dernières sur un lien de communication transatlantique. La seconde

exposition s'est faite à l'INET '96, à Montréal. Les visiteurs de l'INET ont contribué au logiciel par leurs commentaires au niveau des multiples aspects de la navigation avec un gant numérique jumelé au casque d'immersion.

Après avoir pris un certain recul depuis la création de cet outil logiciel, il devient aisé d'avoir un regard critique plus objectif sur le produit final. Certes, *InterWorlds* comporte des lacunes évidentes, mais suite aux commentaires recueillis des participants de ces présentations officielles, cette application de navigation dans des environnements virtuels possède aussi certaines forces indéniables, dont les principales sont les suivantes:

- l'implantation d'un mécanisme d'automatisation de la création d'environnements virtuels utilisant un fichier de configuration comme base de description;
- la consultation de larges banques d'informations tridimensionnelles, autant pour la représentation des données que pour la navigation à travers celles-ci;
- le filtrage par ondelettes des signaux bidimensionnels provenant des périphériques d'immersion assurant une navigation non saccadée et régulière;
- le support de multiples périphériques d'entrée/sortie permettant une immersion plus convaincante pour l'utilisateur;
- malgré ses limites, l'utilisation d'un gant pour la navigation permet des déplacements plus naturels que tout autre périphérique bidimensionnel (une souris, par exemple);
- la consultation des séquences vidéo s'effectue à même l'environnement virtuel, sans exiger du participant qu'il termine sa navigation pour effectuer ce visionnage.

Quant aux faiblesses, les commentaires négatifs et les recommandations formulées par les utilisateurs d'*InterWorlds* peuvent être regroupés en trois catégories différentes :

- l'absence de support pour une immersion virtuelle en réseau;
- L'imposition de périphériques d'immersion peu répandus (le gant et le casque);

- le faible niveau de réalisme de l'environnement ainsi que l'aspect cubique des objets.

Les premières fonctionnalités d'*InterWorlds* ne constituent, bien entendu, qu'un prototype permettant de constater jusqu'à quel point cette méthode de navigation est prometteuse. Les prochaines versions, en plus d'apporter des corrections afin de pallier aux lacunes énumérées précédemment, incluront aussi l'intégration de toutes nouvelles idées. Par exemple, dans les prochaines versions d'*InterWorlds*, il pourrait être intéressant d'ajouter le support des fichiers de type VRML pour éventuellement permettre les immersions à plusieurs utilisateurs dans des mondes sur Internet. Pour ce faire, le développement d'un serveur d'environnements virtuels devrait être complété afin de gérer les interactions et les modifications des utilisateurs répartis dans le monde et de répercuter ces dernières à travers tous les utilisateurs.

Le succès d'*InterWorlds* et les commentaires positifs reçus à son égard permettent de constater que la réalité virtuelle est une alternative très intéressante pour la navigation dans de larges banques d'informations. Il suffirait probablement de combler quelques-unes des lacunes énumérées plus haut pour en faire un puissant outil de recherche.

De plus, l'ajout des fonctionnalités permettant d'interagir avec d'autres participants dans l'environnement virtuel serait aussi un apport majeur. En effet, les tests effectués sur ce logiciel permettent de conclure que la réalité virtuelle semble être, contrairement à l'idée préconçue, une technologie de socialisation, plutôt qu'un environnement hermétique favorisant l'isolement de l'individu.

ANNEXES

Annexe A: Introduction à la réalité virtuelle

Les prochains paragraphes correspondent à une brève description de la réalité virtuelle écrite par Éric Jobidon au début des années 1990. L'objectif principal de ce document était de vulgariser cette nouvelle discipline afin d'en faire une nouvelle alternative d'investissement autant pour le milieu privé que pour certains organismes gouvernementaux. La nature quasi-simpliste des propos de ce document peut s'expliquer par le faible niveau de connaissance technique de l'auditoire cible. Cette contrainte permet par contre une simplification des concepts qui, à son tour, se traduit par un aperçu global assez complet de la réalité virtuelle.

Émergeant à l'origine directement de l'imagination de quelques chercheurs non-conformistes désirant concevoir l'ultime interface humain-machine, la réalité virtuelle est maintenant beaucoup plus qu'un idéal: elle fait maintenant partie de la **réalité**!

Qu'est-ce que la réalité virtuelle?

C'est dans un laboratoire de *Silicon Valley* en Californie, au tout début des années 1970, que le concept de la réalité virtuelle fit son apparition. Malgré un développement technologique très lent s'échelonnant sur plus de 20 ans, ce n'est que tout récemment que ce nouveau médium a franchi le seuil de l'acceptation culturelle lui assurant ainsi une évolution à la fois plus substantielle et plus constante. Ainsi, depuis quelques années, certains géants japonais de l'industrie de la haute technologie tels que Fujitsu et Matsushita de même que d'autres

groupes industriels incluant MCA et NYNEX, conjointement avec des centres de recherche universitaires américains, repoussent les limites technologiques de ce médium nouveau genre.

Définition

Certains spécialistes la décrivent comme «une réalité tridimensionnelle implémentée avec des lunettes stéréoscopiques et des gants d'interface avec la réalité», alors que d'autres parlent plutôt de «mondes alternatifs remplis d'images générées par ordinateur répondant aux mouvements d'un humain»... En fait, on peut décrire la réalité virtuelle autant avec les équipements qu'elle requiert qu'avec les expériences qu'elle rend disponibles.

La réalité virtuelle est, somme toute, une innovation prometteuse sur plusieurs plans. Au plan technique, elle devient un nouveau mode de conception, facilitant le développement de nouveaux produits et procédés. Au plan humain, elle ouvre la porte à de nouvelles techniques de communications alors que d'un point de vue pédagogique, elle permet aux chercheurs de prendre un certain recul face au monde qui nous entoure.

Les vastes possibilités que nous offre la réalité virtuelle nous restreignent à donner une définition générale de ce qu'elle est et de ce qu'elle permet. Une définition plus contraignante viendrait à l'encontre des opportunités illimitées que nous donne ce médium de demain.

Équipement requis

Pour voyager dans le monde de la réalité virtuelle, un équipement sophistiqué est requis. Ce dernier inclut: un casque, doté de deux écrans à cristaux liquides; des gants, permettant la manipulation d'objets virtuels; un habit particulier, donnant à l'ordinateur plus de renseignements sur les déplacements effectués; ainsi qu'un ordinateur, générant ce monde artificiel. Plusieurs accessoires permettent de parfaire cette illusion, mentionnons à titre d'exemples: un tapis roulant, des écouteurs, des détecteurs de mouvements installés dans la pièce informant l'ordinateur de la position exacte du voyageur ou encore des dispositifs mécaniques de "retour d'efforts" donnant ainsi le sens du toucher à même le monde virtuel.

L'équipement disponible aujourd'hui est assez restrictif quant à la nature des mouvements qu'il permet. En fait, l'utilisateur est confiné à une pièce, sans trop pouvoir en sortir, étant donné tous les fils le reliant à l'ordinateur. Avec les futurs développements technologiques, toutes ces contraintes seront amoindries sinon totalement éliminées sous peu, permettant ainsi une immersion totale dans une réalité virtuelle.

Ultimement, dans une vision à long terme, mis à part l'ordinateur lui-même, aucun équipement extérieur ne sera requis pour plonger l'utilisateur dans des voyages virtuels.

La réalité virtuelle, aujourd'hui

Déjà aujourd'hui, on retrouve plusieurs domaines utilisant la réalité virtuelle. Celle-ci permet, entre autres, de faciliter certaines tâches et même de facilement manipuler toutes sortes d'informations (données boursières, liens intramoléculaires, informations textuelles, plans et devis, renseignements démographiques et géographiques, etc.).

L'un des domaines de la haute technologie le plus poussé au Québec est celui des simulateurs de vol. Il s'agit en fait d'un sous-ensemble de la réalité virtuelle. On pourrait effectivement effectuer une analogie entre l'état de la réalité virtuelle d'aujourd'hui et celui des simulateurs de transports aériens tels ceux conçus et fabriqués par CAE, ici même à Montréal. Ces simulateurs offrent la possibilité à de futurs pilotes de vivre des expériences virtuelles leur permettant de se familiariser avec les commandes de leur appareil comme s'ils étaient en plein vol. Assis dans une cabine de pilotage en tous points identique à celle des avions de ligne qu'ils piloteront sous peu, les apprentis voient devant eux dérouler une scène générée par ordinateur représentant des situations de vol particulières. Tout l'environnement du pilote y est reproduit, les turbulences causent de légères secousses qui sont ressenties à même le simulateur, les phénomènes atmosphériques y sont reproduits avec un réalisme impressionnant et même les accélérations/décélérations se traduisent par une force gravitationnelle clouant le pilote à son siège!

Un autre exemple d'utilisation de la réalité virtuelle est le programme conçu par la firme Maxus Systems International qui permet aux gérants d'un fonds de retraite américain (le TIAA-CREF de New-York) de vivre «en direct» et en trois dimensions l'évolution de leurs placements en bourse. Grâce à ce système, ils voient les actions comme des carrés de couleurs sur un grand échiquier. À chaque colonne correspond une place boursière donnée, à chaque rangée un secteur industriel ou de marché. Si une action vient à chuter, la case correspondante vire au rouge. L'opérateur peut alors immédiatement s'informer sur les mouvements en cours et réagir en conséquence.

Mais, plus que tout autre domaine, c'est l'industrie du divertissement qui profite le plus de la venue des technologies de la réalité virtuelle: des dizaines de jeux vidéo ainsi que d'autres destinés aux arcades ont déjà fait leur apparition, principalement chez nos voisins du Sud. Les plus récents permettent à l'utilisateur de vivre une expérience de jeu similaire à la vie au temps du roi Arthur: le «joueur» prend le rôle de Merlin ou celui d'un preux chevalier et explore le vaste monde virtuel qui s'offre à lui.

Ici, au Québec, cette révolution se fait plus doucement: seuls quelques jeux d'arcades sont disponibles étant donné l'ignorance de la majorité de la population sur ce sujet. Ce n'est effectivement qu'une question de temps étant donné l'attrait social pour la haute technologie que partagent les Québécois.

La réalité virtuelle de demain

La réalité virtuelle de demain pourrait en fait englober la quasi totalité des activités quotidiennes d'aujourd'hui. On imagine très bien un citoyen en train de faire son jogging «virtuel» sur la lune, la tenue d'une conférence entre plusieurs filiales d'une compagnie dans des décors exotiques, ou même un journalier qui se lève le matin, enfle le survêtement de réalité virtuelle et effectue sa journée de travail quotidienne chez lui, tout en rencontrant *virtuellement* ses confrères de travail qui se trouvent eux-mêmes à leur propre résidence. En fait, la réalité virtuelle et ses applications ne sont limitées que par l'imagination des individus qui la créent.

Prenons par exemple une compagnie qui doit concevoir un nouveau moteur. La réalité virtuelle permettrait aux ingénieurs de voir le moteur, de pouvoir lui toucher et de le modifier et ce, plusieurs mois avant sa mise en fabrication. De cette façon, avant d'investir temps et argent dans un projet à haut risque, les concepteurs pourraient soumettre le prototype du moteur à une panoplie de tests virtuels, constater les résultats et adapter le moteur instantanément aux faiblesses décelées. Tout ce processus se ferait en manipulant la pièce à travers l'équipement de réalité virtuelle (gants, casque, etc.); l'ordinateur en rectifiant automatiquement les plans.

Le domaine médical en est un qui profitera, sans aucun doute, grandement de l'évolution des techniques de virtualisation. Par exemple, l'accès à un monde microscopique à travers la réalité virtuelle permettra aux pharmacologues la manipulation directe de molécules afin d'accélérer la découverte des résultats recherchés. D'un point de vue psychologique, la réalité virtuelle permettra au psychologue de «guérir» l'enfant autistique en accédant à son monde interne et en le guidant hors de cet univers psychotique.

Parallèlement, on peut imaginer un chirurgien réputé, établi à Montréal, effectuant une opération chirurgicale sur un patient situé en Somalie, par exemple. La téléprésence (utilisation de la réalité virtuelle permettant de contrôler des robots à distance) offrirait au spécialiste la possibilité de diriger un automate afin que celui-ci pratique l'opération sous ses commandes. Le chirurgien verrait la blessure et pourrait diriger le robot afin qu'il donne les soins nécessaires au malade somalien.

De façon similaire, l'architecte pourrait facilement visiter avec les investisseurs un bâtiment avant même que celui-ci soit construit et ainsi lui permettre d'adapter l'édifice aux besoins des acheteurs en «poussant» sur un mur trop long ou en déplaçant une porte afin de la rendre plus accessible. L'ordinateur assisterait l'architecte en modifiant et en sauvegardant les plans de la construction selon ces nouvelles spécifications.

Que retire-t-on en investissant dans ce programme?

L'importance de la réalité virtuelle au Québec

Le Québec dégage une image d'innovateur et de leader mondial principalement dans des domaines de la haute technologie et des télécommunications. Des compagnies telles Bombardier, Téléglobe Canada, Vidéoway et SoftImage, en plus d'être chefs de file dans leur domaine respectif, contribuent à dorer l'image du Canada (et tout particulièrement du Québec) à travers le monde.

Comme nous avons pu le constater précédemment, la réalité virtuelle est un domaine très prometteur: elle permet, entre autres choses, de redéfinir la notion de médium de

communications. Par exemple, au lieu de «regarder» les nouvelles du soir, un téléspectateur pourrait littéralement y participer et les vivre intégralement. En «plongeant» le Québec dans cette direction, on contribuerait à propager l'image d'avant-gardisme nationale et internationale véhiculée par notre province. Il devient donc très opportun d'investir dans un tel domaine de haute technologie.

Au point de vue sociétair, on devrait promouvoir l'émergence de nouvelles technologies et les développer ici même, au Québec. La mise sur pied d'un centre de recherche sur la réalité virtuelle permettrait non seulement la création d'emplois, mais aussi d'orienter la recherche et le développement dans ce domaine. Ce centre offrirait des services de consultation, publierait des ouvrages précisant le développement effectué afin de tenir la population informée sur le sujet, servirait de référence pour les compagnies voulant utiliser les bienfaits de ce domaine et pourrait même offrir des stages pour les universités québécoises ou canadiennes. En fait, la R&D effectuée par le centre deviendrait un investissement profitable dans la formation de la main-d'oeuvre de demain.

Une nouvelle méthodologie de travail

En plus de l'aspect innovateur, la RV apporte une nouvelle méthodologie de travail et ce, dans plusieurs domaines. Par exemple, la création de nouveaux concepts serait facilitée par des outils de réalité virtuelle, la réparation de pièces d'équipement serait accélérée et moins dangereuse grâce à la téléprésence. En enseignement, cette technologie procurerait certainement de nouveaux outils pédagogiques ainsi qu'une motivation additionnelle aux élèves de tous les âges. Voilà tous des exemples où la réalité virtuelle s'avère une méthode plus efficace, rapide, plus performante et aussi plus précise que les méthodes traditionnelles.

Les programmes universitaires de second cycle

Étant donné que la réalité virtuelle en est à ses premiers balbutiements, aucun établissement postsecondaire n'offre de formation dans ce domaine. Même si la réalité virtuelle s'apparente à la vision par ordinateur et au graphique tridimensionnel sur ordinateur (deux disciplines offertes à travers le vaste réseau d'universités canadiennes), une grande différence les sépare tout de même. En fait, alors que les universités canadiennes sont toujours sous équipées pour offrir un tel programme, ce n'est que depuis quelques années que certaines universités américaines offrent des diplômes en réalité virtuelle.

Depuis maintenant trois ans, le laboratoire des technologies d'interfaces avec l'humain (*Human Interface Technologies Lab*) de l'Université de Washington donne accès à des programmes de deuxième et troisième cycle offrant une formation poussée dans le domaine de

la réalité virtuelle. Le HITLab est une des institutions les mieux cotées dans le domaine, avec l'Université de Caroline du Nord (campus de Chapel Hill).

La notoriété de ce laboratoire se traduit par une implication internationale continue dans le domaine de la réalité virtuelle. La création de la «*Virtual Worlds Society*», une organisation éducative internationale à but non lucratif ayant pour mission le développement du VR (*Virtual Reality*), en est un bel exemple.

Le programme universitaire du HITLab a une durée de 2 ans et exige de la part des étudiants que la totalité des frais scolaires soit disponible avant le début des études. Un montant de US\$22 500 par année est exigé et couvre la totalité des dépenses encourues incluant la période estivale. Ce montant couvre tant les frais d'inscriptions que la nourriture et le logement pendant l'année scolaire.

Conclusion

Avec une formation aussi poussée au HITLab de l'Université de Washington, je pourrais effectuer des études pour certains organismes concernant la réalisation de tests de faisabilité et d'implantation de technologies de réalité virtuelle. Des emplois d'été dans les entreprises commanditaires deviennent aussi possibles, de même que la mise sur pied d'un centre de recherche sur cette nouvelle technologie avant-gardiste.

Annexe B: Filtrage par ondelettes

Les pages qui suivent contiennent le code source en C qui a permis d'incorporer la transformée ondelette rapide dans le logiciel de navigation pour l'environnement virtuel. Les fonctions dont le nom débute avec les lettres **WT_** font appel à des routines d'un noyau graphique. Celles qui ne sont pas énumérées ici font référence à des fonctions de la librairie *WaveLib*. Étant donné la nature publique de celles-ci, ni le code, ni la documentation de ses fonctions ne sera fourni ici. Par contre, sur demande, celles-ci pourront être résumées en faisant une demande directement à l'auteur.

```
//-----Variables et #defines globaux-----
BOOL PRINT_IS_ON, COMPRESSION_IS_ON;

#define WAVELET_ARRAY_SIZE 16
#define WAVELET_TRESHOLD 30

typedef struct _wavelets_data
{
    int indice; //L'indice courant dans la liste circulaire
    Wtp3 raw_values[WAVELET_ARRAY_SIZE]; // Vecteur des valeurs brutes
    Wtp3 filtered_values[WAVELET_ARRAY_SIZE]; // Vecteur des valeurs filtrées
} wavelets_data;

//variable globale statique nécessaire pour la persistance des données
// du tracker wavelets_data tdata;

#define sign(num) (((num)>0.0)?1.0 :-1.0)

//-----

/*****
 *
 * La fonction Treshold permet le filtrage proprement dit des valeurs.
 * Les coefficients de l'ondelette transformés reçus ici sont filtrés selon
 * une prémisses de base : si x < WAVELET_TRESHOLD → x = 0. En effet, si
 * les coefficients sont inférieurs à ce seuil, ils ne contiennent
 * probablement pas l'énergie du signal original. Il s'agit plutôt de bruit.
 * On peut se départir du bruit en les forçant à zéro de cette façon.
 *
 *****/
```



```

* Accepte en entrée un pointeur sur un vecteur transformé et effectue
* directement sur ce vecteur le thresholding.
* Aucune sortie
*****/

void Threshold(VEC *v)
{
    u_int i;

    if (( v == (VEC *)NULL ) || ( v->ve == (Real *)NULL ))
        return;

    for ( i=0; i<v->dim; i++)
    {
        //utiliser une fonction de «soft thresholding» comme dans Wavelets for kids
        v->ve[i] = sign(v->ve[i]) * ( fabs(v->ve[i]) - (sqrt(log(v->dim)))/(sqrt(v->dim))) );

        // et/ou tous les coefficients inférieurs à WAVELET_TRESHOLD sont fixés à 0
        v->ve[i] = (fabs(v->ve[i]) < WAVELET_TRESHOLD )? 0.0 : v->ve[i];
    }
}

/*****
*
* La fonction MakeSample crée un vecteur d'échantillons en utilisant
* les coefficients reçus du tracker.
* La valeur la plus récente est entreposée à la "fin" du vecteur
* circulaire en utilisant un indice de position. En optant pour cette
* technique, on peut utiliser les données précédentes lors de la
* décomposition ondelette.
*
*****/

VEC *MakeSample(Wtp3 vect[], int indice, int which)
{
    VEC *x=v_get(WAVELET_ARRAY_SIZE);
    int i;

    /*
    * La donnée à vect[indice] correspond à la dernière info reçue du tracker. Dans le
    * vecteur de sortie, x, on stocke à l'indice 0 la plus vieille donnée du tracker
    * (vect[indice+1]). Ceci afin que les bornes n'affectent pas les wavelets.
    * En fait, on tente de traiter l'info la plus récente en fonction du plus grand
    * nombre de valeurs connues possible.
    * Ex. : si vect=[5,7,9,3] et que l'indice=2 alors les données ont été reçues dans
    * l'ordre3, 5, 7, 9. Le vecteur x contiendra cet ordonnancement.
    */

    for(i=0;i<WAVELET_ARRAY_SIZE;i++)
    {
        x->ve[i]=vect[(i+indice+1)%WAVELET_ARRAY_SIZE][which];
    }

    return x;
}

/*****
*
* La fonction Filter filtre les données reçues en paramètre.
* Les coefficients transformés reçus ici sont filtrés selon la
* prémisse de base suivante : si  $x < 1,2 \rightarrow x = 0$ 
*
* Accepte en entrée un pointeur sur la structure contenant les données
* données du tracker et agit directement sur ce vecteur.

```

```

* Aucune sortie
*
*****/

void Filter(wavelets_data *tdata)
{
    VEC *x=NULL,*xr=NULL;
    FILTER *ga=NULL,*gs=NULL, *ha=NULL, *hs=NULL;
    Real a,tol;
    int L, i, aa, which = X;
    char s;
    COMPRESS *w;

    //Valeur initiale des variables globales
    PRINT_IS_ON=FALSE;
    COMPRESSION_IS_ON=FALSE;

    //Création du filtre
    daubechies(&ha,3); //3=nbre de moments nuls
    hs=copyfilter(NULL,ha);
    ga=bimod(hs);
    gs=bimod(ha);

    for (which=X; which<=Z; which++) //faire le même traitement pour pitch, roll et yaw
    {
        //Création de l'échantillon

        // [2^(L=4)]=16 données dans le signal en entrée
        x=MakeSample(tdata->raw_values, tdata->indice, which); //création des échantillons
        printf("\nValeurs du signal (+ bruit) en entrée :\n");
        v_output(x);

        //Transformation du signal d'entrée avec fwt selon
        // des paramètres précis
        tol= 4; //Fixe la Tolerance.
        L = 4; //Détermine le nombre de niveaux
        w=compress1(ha,ga,x,L,tol);

        //Impression des coefficients transformés
        v_output((VEC *) (w->cpe));

        //Tresholding des coefficients
        Treshold((VEC *) (w->cpe));

        //Impression des coefficients transformés et tresholdés
        v_output((VEC *) (w->cpe));

        //Décompression des coefficients
        xr=uncompress1(hs,gs,w);

        //Copie des valeurs dans tdata
        /*
        * Toutes les valeurs du vecteur sont copiées. Ainsi, les nouvelles valeurs
        * filtrées par les wavelets remplaceront celles déjà calculées. Ceci stabilise
        * davantage le casque puisqu'on tente de filtrer en se fiant à des valeurs
        * déjà filtrées.
        */
        aa=tdata->indice+1;
        for(i=0;i<WAVELET_ARRAY_SIZE;i++,aa++)
            tdata->filtered_values[aa*WAVELET_ARRAY_SIZE][which] = x->ve[i];

        //Affichage du signal filtré
        v_output(xr);

        //Affichage des statistiques du traitement
        v_results(x,xr);

        //Clean up local...
        v_free(x);
        compress_free(w);
        v_free(xr);
    }
}

```

```

    }

    //Clean up régional
    f_free(ha);
    f_free(hs);
    f_free(ga);
    f_free(gs);
}

/*****
*
* La fonction HeadTrackerUpdateFonction est un module contrôlant
* comment le tracker fournit des données d'état.
*
*****/
void HeadTrackerUpdateFonction(WTsensord *t)
{
    WTiglasses_rawdata *raw;
    WTp3 p, filtered_p;
    WTq q, raw_q;

    //Mise à jour matérielle des informations brutes :
    //Demande de données du tracker
    WTiglasses_rawupdate(t);

    //Attendre que le tracker reçoive la requête
    WTmsleep(25);

    //lire les données retournées par le tracker
    raw=(WTiglasses_rawdata *) WTsensor_getrawdata(t);

    //Ajuster l'indice des listes circulaires
    tdata.indice = (tdata.indice +1) % WAVELET_ARRAY_SIZE;
    WTp3_copy(raw->e, tdata.raw_values[tdata.indice]);

    //Filtrage des données
    Filter(&tdata);

    WTp3_copy(tdata.filtered_values[tdata.indice], filtered_p);
    WTeuler_2q(filtered_p[X], filtered_p[Y], filtered_p[Z], raw_q);

    //relativiser les données du casque avec celles reçues précédemment
    WTsensor_relativizerecord(t, filtered_p, raw_q, p, q);

    //sauvegarder données pour relativisation lors du prochain appel
    WTsensor_setrecord(t, p, q);
    WTsensor_setlastrecord(t, filtered_p, raw_q);
}

```

Annexe C: L'Updatefunction pour le gant

Le code source figurant dans cette annexe représente la fonction en C permettant de faire correspondre les mouvements physiques de la tête du participant avec l'image qu'il a de l'environnement synthétique. À l'intérieur de l'application finale, cette fonction est invoquée périodiquement et c'est à elle que revient la tâche de faire la lecture du statut du casque d'immersion (via un appel à la fonction `WTiglasses_rawupdate(t);`). En plus de cette mise à jour des données, la fonction doit relativiser les valeurs brutes récupérées du capteur de mouvement et, finalement, filtrer les données obtenues à l'aide d'une fonction de filtre (dans ce cas-ci, la transformée ondelette rapide, décrite à l'annexe B).

```
#define WAVELET_ARRAY_SIZE 16
typedef struct _wavelets_data
{
    int indice;
    WTP3 raw_values[WAVELET_ARRAY_SIZE];
    WTP3 filtered_values[WAVELET_ARRAY_SIZE];
} wavelets_data;

wavelets_data tdata;

/*****
*
* La fonction myupdatefn se charge de fournir les valeurs relativisées du
* capteur de mouvement du casque d'immersion. Un certain filtrage des
* données est effectué afin de stabiliser le signal reçu.
*
* Accepte en entrée un pointeur sur l'objet WTsensord correspondant, dans
* ce cas-ci, au capteur de mouvement du casque d'immersion.
* Aucune sortie
*
*****/

void myupdatefn(WTsensord *t)
{
    WTiglasses_rawdata *raw;
    WTP3 p, filtered_p;
    WTP3 q, raw_q;
    wavelets_data *tracker_data;    // nécessaire pour le filtrage par ondelettes

    // Obtention des données du capteur de mouvements.
```

```

// L'appel de cette fonction ne fait qu'indiquer au matériel qu'une mise à jour
// est requise. Les données brutes sont récupérées avec un appel à
// WTsensord_getrawdata(t);
WTiglasses_rawupdate(t);

// Permettre au matériel (capteur de mouvements) de traiter la requête et de
// préparer les valeurs à retourner
WTmsleep(25);

// Récupération proprement dite des valeurs du capteur de mouvement.
raw=(WTiglasses_rawdata *) WTsensord_getrawdata(t);

// Traitement requis pour effectuer le filtrage
tracker_data = (wavelets_data *)WTsensord_getdata(t);
tracker_data->indice = (tracker_data->indice +1) % WAVELET_ARRAY_SIZE;
WTp3_copy(raw->e, tracker_data->raw_values[tracker_data->indice]);

// Filtrage des valeurs
Filter(tracker_data);

//le setdata n'est pas requis puisque l'espace mémoire est global et que tracker_data
// y pointe directement, en modifiant son contenu. Le setdata ferait "perdre"
// l'adresse en mémoire ou l'espace est déjà réservé
// WTsensord_setdata(t, (void *)tracker_data);

// Récupération des valeurs filtrées
WTp3_copy(tracker_data->filtered_values[tracker_data->indice], filtered_p);
WTeuler_2q(filtered_p[X], filtered_p[Y], filtered_p[Z], raw_q);

// Relativisation des valeurs
WTsensord_relativizerecord(t, filtered_p, raw_q, p, q);
WTsensord_setrecord(t, p, q);

// Impression des différences entre les coefficients bruts et les valeurs filtrées
printf("Tracker data:  % 16.8f  % 16.8f  % 16.8f.\r",
        filtered_p[X]-raw->e[X],
        filtered_p[Y]-raw->e[Y],
        filtered_p[Z]-raw->e[Z]);

// Les données finales sont rendues publiques, permettant ainsi aux appels ultérieurs
// de WTsensord_relativizerecord() d'y avoir accès.
WTsensord_setlastrecord(t, filtered_p, raw_q);
}

```

Annexe D: Abstraction des états du gant

Le code source récupéré ci-dessous permet de mieux saisir le fonctionnement des routines intervenant dans l'interprétation et la décomposition des mouvements de la main. Après la définition de quelques variables globales, le corps de la fonction **WT5glove_interpret** est retranscrit. Cette dernière se charge d'interpréter la quantité de flexion des doigts du gant numérique et de déterminer quel geste l'utilisateur tente de produire. L'interaction avec les autres fonctions de l'application se fait alors par le biais de variables globales.

```
#define Constrain(x,a,b) if (x<a) x=a; else if (x>b) x=b;

#define R90      (3.141592653/256.0)

/*Définition des trois états possibles pour chaque doigt: ouvert (Open), Fermé (Closed)
 * ou Pas important (Don't Care). Chacun des doigt du gant peut être soit ouvert (en
 * pleine extension) ou fermé (une flexion complète). La troisième valeur permet de
 * faire abstraction de l'état d'un doigt.
 */
#define OPEN      0
#define CLOSED    1
#define DC        2

/* Définition des gestes les plus utilisés. Chaque ligne de cet ensemble représente un
 * geste distinct pour le participant. Chacun des geste est défini en fonction de
 * l'état de chacun des doigts du gant: soit OPEN, CLOSED ou DC. La première valeur
 * correspond à l'état du pouce puis, dans l'ordre, les suivantes correspondent à
 * l'état de l'index, du majeur, de l'annulaire et de l'auriculaire.
 */
#define MAXGEST 6
char WT5glove_gesttabl[MAXGEST][5]={
    {CLOSED,OPEN,CLOSED,CLOSED,CLOSED},    // pointer avec un doigt =0
    {CLOSED,OPEN,OPEN,CLOSED,CLOSED},      // pointer avec deux doigts =1
    {CLOSED,CLOSED,OPEN,OPEN,OPEN},        // pouce et index collés =2
    {OPEN,OPEN,CLOSED,CLOSED,CLOSED},      // fusil =3
    {CLOSED,CLOSED,CLOSED,CLOSED,CLOSED},  // poing =4
    {OPEN,OPEN,OPEN,OPEN,OPEN} };          // main à plat =5
```

```

char WT5glove_state = OPEN; // last latching gesture was open

// Vecteur des valeurs de seuillage pour chacun des doigts
char WT5glove_thresh[5] = {(char)128, (char)128, (char)128, (char)128, (char)128};

static char WT5glove_timeout=3;

// Par défaut, le dernier geste reconnu est la main à plat.
static char WT5glove_lastgest=5;

WTpq WT5glove_handpq={ {0.0, 0.0, ((float)ZMAX+(float)ZMIN)/2.0}, {0.0,0.0,0.0,1.0} };
int moving_dir; //0= pas de mouvement, +1= avancer, -1= reculer

/*****
*
* La fonction WT5glove_interpret se charge d'interpréter les valeurs de
* quantité de flexion des doigts du gant numérique. Les valeurs brutes de
* chacun des doigts lui sont fournies en entrée et, selon la définition
* des différents gestes, des variables globales de mouvement et de
* direction sont mises à jour.
*
* Accepte en entrée un pointeur sur l'objet WTsensor correspondant, dans
* ce cas-ci, au capteur de mouvement du casque d'immersion ainsi qu'un
* second pointeur sur la structure contenant les données représentant le
* degré de flexion de chacun des doigt du gant.
* Aucune sortie
*
*****/

void WT5glove_interpret (WTsensor *sensor, WT5glove_data *dt)
{
    // figure gestures out
    int i;
    float as, ts, wx, wy, wz;
    WTpq l6d;
    WTq q1,q2;

    /*
    * Pour chacun des gestes prédéfinis, vérifier si les données en entrée
    * correspondent à la quantité de flexion définie pour ce geste.
    */
    for (i=0; i<MAXGEST; i++)
    {
        char yes=1;
        int j;

        // Pour chaque doigt de la main, effectuer les tests de seuillage
        for (j=0; j<5; j++) switch (WT5glove_gesttabl[i][j])
        {
            case DC:
                break; // dont care = match always

            case OPEN:
                // Pour que le doigt soit considéré ouvert, il faut que la quantité
                // de flexion soit inférieure au seuil défini dans WT5glove_thresh.
                yes &= (char)(dt->bFinger[j]<WT5glove_thresh[j]);
                break;

            case CLOSED:
                // Pour que le doigt soit considéré ouvert, il faut que la quantité
                // de flexion soit supérieure au seuil défini dans WT5glove_thresh.
                yes &= (char)(dt->bFinger[j]>WT5glove_thresh[j]);
                break;
        }
        if (yes) goto found_it;
    }
    WT5glove_timeout = 5;
    return;
}

```

```

found_it:          // la position des doigts actuelle correspond au geste i
wx = wy = wz = 0.0;
l6d.p[X] = l6d.p[Y] = l6d.p[Z] = 0.0;

if (i==WT5glove_lastgest)
{
    // il ne faut pas diminuer le timeout si celui-ci est relativement grand
    if (WT5glove_timeout>100)
        goto new_data

    // diminuer le timeout pour que l'action engendrée par le gant ait le
    // temps de s'exécuter avant qu'un autre geste ne soit détecté.
    WT5glove_timeout--;

    if (WT5glove_timeout)
        goto new_data; // c'est donc un nouveau geste
}
else
{
    WT5glove_timeout = 3;
    WT5glove_lastgest = i;    // nouveau geste
    goto new_data;
}

ts=WTsensor_getsensitivity(sensor)/128.0;
as=WTsensor_getangularrate(sensor)/128.0;

// déterminer le type du nouveau geste effectué
switch (i)
{
    case 0:
    {
        // pointer avec un seul doigt = avancer
        SelectingObject = FALSE;

        l6d.p[Z] = ts*(dt->bPitch-128);
        //printf("roll= %03x  \r", dt->bRoll);
        l6d.p[X] = (dt->bRoll)/-30.0;
        l6d.p[Z] /= -40.0;
        wy /= -40.0;

        // Accélération?
        // WT5glove_handpq.p[Y] = SPEED * ts * dt->bPitch-128);
        WT5glove_handpq.p[Y] = (float)(dt->bPitch-128);
        WT5glove_handpq.p[X] = (float)(dt->bRoll-128);
        moving_dir = 1;        // Avancer
        WT5glove_timeout=1;    // repeat gesture
        break;
    }

    case 1:
    {
        //2 doigts = reculer

        SelectingObject = FALSE;
        l6d.p[Z] = ts*(dt->bPitch-128);
        wy = as*(dt->bRoll-128);
        l6d.p[Z] /= 40.0;
        wy /= 40.0;

        WT5glove_handpq.p[Y] = (float)(dt->bPitch-128);
        WT5glove_handpq.p[X] = (float)(dt->bRoll-128);
        moving_dir = -1;
        WT5glove_timeout = 1; // repeat gesture
        break;
    }

    case 2:
    {
        //3 doigts = STOP video
        stop_video();
    }
}

```



```

        SelectingObject = FALSE;
        moving_dir = 0;
        WT5glove_timeout = (char)250;          // do not repeat gesture

        // Ajouter un facteur d'accélération?
        /* WT5glove_handpq.p[Z] += SPEED*ts*(dt->bPitch-128);
           WT5glove_handpq.p[X] += SPEED*ts*(dt->bRoll-128);
           WT5glove_timeout = 1; // repeat gesture
        */
        break;
    }

    case 3:
    {
        // gun = Select object (START video, hear sound, etc.)

        SelectingObject = TRUE;
        moving_dir = 0;
        WT5glove_timeout = (char)250; // do not repeat gesture
        break;
    }

    case 4:
    {
        // fist=Do nothing
        SelectingObject = FALSE;
        WT5glove_timeout = (char)250; // do not repeat gesture
        moving_dir = 0;
        break;
    }

    case 5:
    {
        // flat=move up/down
        SelectingObject = FALSE;
        moving_dir = 0;
        WT5glove_timeout = (char)250; // do not repeat gesture
        break;
    }
} //switch(i)

new_data:
    WTeuler_2q (wx,wy,wz,16d.q);
    WTsensor_setrecord (sensor,16d.p,16d.q);

    Constrain (WT5glove_handpq.p[X],(float)-XYEXT,(float)XYEXT);
    Constrain (WT5glove_handpq.p[Y],(float)-XYEXT,(float)XYEXT);
    Constrain (WT5glove_handpq.p[Z],(float)ZMIN,(float)ZMAX);

    WTeuler_2q ((dt->bPitch)*R90,0.0,(dt->bRoll)*R90,WT5glove_handpq.q);
}

```

Annexe E: Fichier de configuration du monde

Les prochaines lignes sont directement tirées d'un fichier de configuration de l'environnement virtuel. Le fichier est décomposé en de multiples blocs, chacun identifié par un titre entouré d'accolades. Par exemple, la première section est titrée *[Lobby]* et ne contient qu'une seule ligne: *NFF=Living, Bureau*. Ainsi, chaque ligne d'une section est composée d'un libellé (ici, *NFF*) ainsi que d'une valeur (*Living, Bureau*). De cette façon, chaque section permet de décrire une partie de l'environnement virtuel.

Le point d'entrée du fichier est la section qui porte le nom de *[Lobby]* et elle contient les différents modules du monde virtuel. Les valeurs *Living* et *Bureau* font référence à des sections du fichier qui serviront à peupler le monde synthétique d'un certain contenu. Dans cet exemple, la section *[Living]* indique qu'un fichier particulier (*/users/share/wtknt/models/radmodel/villa1.NFF*) contient une définition tridimensionnelle d'une pièce. Cette dernière devra subir une rotation de 16 degrés autour de l'axe des X (*ROTX*), de 0 degrés autour de l'axe des Y (*ROTY*) et que toutes les dimensions y figurant devront être décuplées par un facteur de 10. Une fois la pièce créée, des objets viendront meubler la pièce. Ceux-ci sont, encore une fois, définis dans une sous-section du même fichier de configuration.

En utilisant cette forme de hiérarchie, la création d'un espace virtuel devient pratiquement triviale. De **requis**, les connaissances en géométrie tridimensionnelle passent à **complémentaires**, rendant ainsi la tâche de créer un tel environnement, une activité accessible à tous.

```
[Lobby]
NFF=Living, Bureau
```

```
[die]
def=die.NFF
objets=cadre1
```

```
[Living]
def=/users/share/wtknt/models/radmodel/villa1.NFF
objets=cadre1, cadre2, cadre3, cadre4
ROTX=16
ROTY=0
X=2090
SCALE=10
```

```
[Bureau]
def=/users/share/wtknt/models/radmodel/flat1.NFF
objets=cadre5, cadre6, cadre7, cadre8, cadre9, cadre10
ROTY=94
x=1000
Y=1050
Z=770
SCALE=10
```

```
[cadre1]
Type=MPEG_NFF
Image=d:\users\job\fleuve.tga
Lien=d:\users\share\mpeg\fleuve.mpg
X=2000
Y=1005
Z=620
SCALE=2
ROT=2
```

```
[cadre2]
Type=MPEG_NFF
Image=d:\users\job\crac.tga
Lien=d:\users\share\mpeg\crac.mpg
X=2127
Y=1070
Z=1250
SCALE=3
ROT=0
```

```
[cadre3]
Type=MPEG_NFF
Image=d:\users\job\bach-p.tga
Lien=d:\users\share\mpeg\arbres.mpg
X=2565
Y=1010
Z=830
SCALE=3
ROT=0
```

```
[cadre4]
Type=MPEG_NFF
Image=d:\users\job\gounod-p.tga
Lien=d:\users\share\mpeg\gounod.mpg
X=1629
Y=1015
Z=940
scale=2
ROT=4
```

```
[cadre5]
Type=MPEG_NFF
Image=d:\users\job\nic-c.tga
Lien=d:\users\share\mpeg\nicpic.mpg
X=1051
Y=1015
Z=1100
SCALE=3
ROT=4

[cadre6]
Type=MPEG_NFF
Image=d:\users\job\beeth-p.tga
Lien=d:\users\share\mpeg\beethov.mpg
X=305
Y=1015
Z=1050
SCALE=3
ROT=4

[cadre7]
Type=MPEG_NFF
Image=d:\users\job\mauritan.tga
Lien=d:\users\share\mpeg\mauritan.mpg
X=317
Y=1015
Z=500
SCALE=3
ROT=4

[cadre8]
Type=MPEG_NFF
Image=d:\users\job\phar.tga
Lien=d:\users\share\mpeg\pharaons.mpg
X=720
Y=1015
Z=400
SCALE=3
ROT=4

[cadre9]
Type=MPEG_NFF
Image=d:\users\job\scully.tga
Lien=d:\users\share\mpeg\scully.mpg
X=1180
Y=1015
Z=400
SCALE=3
ROT=4

[cadre10]
Type=MPEG_NFF
Image=d:\users\job\nic-c.tga
Lien=d:\users\share\mpeg\nicpic2.mpg
X=770
Y=1010
Z=1060
SCALE=3
ROT=0
```

Annexe F: Exemple de fichier VRML

Les prochaines lignes décrivent le contenu d'un fichier VRML de description d'un environnement virtuel. Dernièrement, plusieurs outils de création et de manipulation des volumes permettent de sauvegarder ou d'exporter une scène tridimensionnelle vers un fichier de format VRML. Un tel fichier est composé de matrices de transformations qui doivent être appliquées à des ensembles de coordonnées dans l'espace. De plus amples détails sur le format des fichiers VRML sont donnés en [25].

```
#VRML V1.0 ascii

# created by Caligari Corp. Fountain (tm)
# copyright 1995 Caligari Corporation

Separator {
  TransformSeparator {
    MatrixTransform {
      matrix
        1.00 0.00 0.00 0
        0.00 0.00 -1.00 0
        0.00 1.00 0.00 0
        4.83 4.09 1.12 1
    }
    DEF LocLight PointLight {
      color 1.00 0.65 0.38
      on TRUE
      location 0.00 0.00 0.00
      intensity 1.30
    }
  }
  TransformSeparator {
    MatrixTransform {
      matrix
        1.00 0.00 0.00 0
        0.00 0.00 -1.00 0
        0.00 1.00 0.00 0
        -3.63 3.05 5.12 1
    }
    DEF LocLight_1 PointLight {
```

```

        color 0.01 0.70 1.00
        on TRUE
        location 0.00 0.00 0.00
        intensity 1.30
    }
}
TransformSeparator {
    MatrixTransform {
        matrix
        1.00 0.00 0.00 0
        0.00 0.00 -1.00 0
        0.00 1.00 0.00 0
        1.23 2.93 -5.39 1
    }
    DEF LocLight_2 PointLight {
        color 0.80 0.80 0.80
        on TRUE
        location 0.00 0.00 0.00
        intensity 1.30
    }
}
Separator {
    MaterialBinding {
        value OVERALL
    }
    Material {
        ambientColor [
            0.10 0.10 0.10,
        ]
        diffuseColor [
            0.82 0.82 0.82,
        ]
        specularColor [
            0.10 0.10 0.10,
        ]
        emissiveColor [
            0.00 0.00 0.00,
        ]
        shininess [
            0.00,
        ]
        transparency [
            0.00,
        ]
    }
    MatrixTransform {
        matrix
        1.00 0.00 0.00 0
        0.00 0.00 -1.00 0
        0.00 1.00 0.00 0
        0.00 1.00 0.00 1
    }
    Coordinate3 {
        point [
            -1.00 -1.00 -1.00,
            -1.00 -1.00 1.00,
            1.00 -1.00 -1.00,
            1.00 -1.00 1.00,
            -1.00 1.00 -1.00,
            1.00 1.00 -1.00,
            1.00 1.00 1.00,
            -1.00 1.00 1.00,
        ]
    }
    ShapeHints {
        creaseAngle 0.5
        vertexOrdering COUNTERCLOCKWISE
        shapeType SOLID
        faceType UNKNOWN_FACE_TYPE
    }
    DEF Cube IndexedFaceSet {
        coordIndex [

```

```
2, 0, 1, 3,  
-1, 4, 0, 2,  
5, -1, 5, 2,  
3, 6, -1, 6,  
3, 1, 7, -1,  
4, 5, 6, 7,  
-1, 0, 4, 7,  
1, -1,  }  
    }  
}
```

BIBLIOGRAPHIE

- [1] J. Vince, "Virtual Reality Systems," Addison-Wesley, 1995.
- [2] W. Bricken, "Virtual Reality: Directions of Growth," tirée des notes SIGGRAPH '90 Panel, 1990.
- [3] M. J. Ackerman, "Medicine Meets Virtual Reality II: Interactive Technology & Healthcare: Visionary Applications for Simulation Visualization Robotics," Aligned Management Associates, San Diego, CA, pp. 5-7, 1994.
- [4] M. Billinghurst, J. Savage, P. Oppenheimer, C. Edmond, "The Expert Surgical Assistant: An Intelligent Virtual Environment with Multimodal Input," HITLab publications, Seattle, WA, 1995.
- [5] R. S. Askew, M. A. Diftler, "Ground Control Testbed for Space Station Freedom Robot Manipulator," IEEE Virtual Reality Annual International Symposium, Seattle, WA, pp. 69-78, septembre 1993.
- [6] P. DuPont. "Overview of Virtual Reality Applications," TC 5 WG 5.10 Workshop on Virtual Environments, Coimbra, Portugal, octobre 1994.
- [7] R. B. Araujo, C. Kirner, "Network Requirements for Large Distributed Virtual Environments," TC 5 WG 5.10 Workshop on Virtual Environments, Coimbra, Portugal, Octobre 1994.
- [8] J. M. Pullen, "Networking for Distributed Virtual Simulation", Computer Networks and ISDN Systems, Vol. 27, 1994.
- [9] D. Edson, "Get World-Class Noise and Total Joy from Your Games with DirectSound and DirectInput," Microsoft Systems Journal, février 1996.
- [10] C. Carlsson, O. Hagsand, "DIVE - a Multi User Virtual Reality System," IEEE Virtual Reality Annual International Symposium, Seattle, WA, pp. 394-400, septembre 1993.
- [11] W. J. King, "Defining Phenomena for an Emotional State Model in the Human Interface," HITLab publications, Seattle, WA, 1994.
- [12] M. Hall, M. Towfiq, G. Arnold, D. Treadwell, H. Sanders, "Windows Sockets: An Open Interface for Network Programming under Microsoft Windows," janvier 1993.
- [13] T. A. Furness III, "Putting Humans into Virtual Space," HITLab publications, Seattle, WA, 1986.
- [14] M. Draper, "Can Your Eyes Make You Sick? Investigating the Relationship Between the Vestibulo-Ocular Reflex and Virtual Reality," HITLab publications, Seattle, WA, 1996.

- [15] R. Hjelsvold, R. Midtstraum, O. Sandsta. "Searching and Browsing a Shared Video Database," Proceedings of the *International Workshop on Multi-Media Database Managment Systems*, Blue Mountain Lake, NY, pp. 90-98, 1995.
- [16] D. A. Keim, H.P. Kriegel, T. Seidl, "Visual Feedback in Querying Large Databases," Proceedings de *Visualization '93*, 1993.
- [17] B. Sawyer, "The ultimate Game Developer's Sourcebook," Coriolis Group Books, pp. 120-124, 1996.
- [18] Y. Meyer, "Wavelets : Algorithms And Applications," Society for Industrial Applied Mathematics, Philadelphie, pp. 13-31, 101-105 , 1993.
- [19] B. Jawerth, W. Sweldens, "An overview of Wavelet Based Multiresolution Analyses," Preprint, p. 19.
- [20] R. A. Devore, B. J. Lucier, "Wavelets ," Preprint, p. 24-28.
- [21] L. Reissell, T. D. DeRose, M. Lounsbery, "Curves And Surfaces," dans les notes *Wavelets and Their Applications in Computer Graphics*, pp. 128-134, 1994.
- [22] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Communications on Pure and Applied Mathematics XLI*, pp. 909-996, 1988.
- [23] M. V. Wickerhausser, "Lectures on Wavelet Packet Algorithms," novembre 1991.
- [24] B. Vidakovic, p. Müller, "Wavelets for Kids : A Tutorial Introduction," Preprint, p. 16, 1991.
- [25] S. N. Matsuba, B. Roehl. "Special Edition Using VRML," QUE, 1996.

